(2)

AFIT/GLM/LSG/88S-3

A DATABASE MANAGEMENT SYSTEM APPLICATION FOR THE GRADUATE PROGRAMS OFFICE OF THE SCHOOL OF SYSTEMS AND LOGISTICS VOLUME 1: DEVELOPMENT AND USER'S MANUAL

THESIS

Phillip H. Beard
Captain, USAF

AFIT/GLM/LSG/88S-3

Approved for public release; distribution unlimited

The contents of the document are technically accurate, and no sensitive items, detrimental ideas, or deleterious information is contained therein. Furthermore, the views expressed in the document are those of the author and do not necessarily reflect the views of the School of Systems and Logistics, the Air University, the United States Air Force, or the Department of Defense.

AFIT/GLM/LSG/88S-3

A DATABASE MANAGEMENT SYSTEM APPLICATION FOR THE GRADUATE PROGRAMS OFFICE OF THE SCHOOL OF SYSTEMS AND LOGISTICS VOLUME 1: DEVELOPMENT AND USER'S MANUAL

THESIS

Presented to the Faculty of the School of Systems and Logistics
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Logistics Management

Phillip H. Beard, B.B.A.
Captain, USAF

September 1988

Preface

The purpose of this study was to develop an efficient computer-based database management system (DBMS) application to automate manual information processing procedures used by the Air Force Institute of Technology School of Systems and Logistics Graduate Programs Office. The Graduate Programs Office had the material resources to develop an automated data processing and information management program but did not have an application or the personnel resources to develop one. This application only considers Graduate Programs Office needs and does not consider AFIT School of Engineering needs. With some minor structural modification, this application could meet School of Engineering needs.

I would like to thank Lt Col Fred Westfall for suggesting this project as my contribution to academia and, in his capacity as my thesis and academic advisor, for his motivational comments such as, "Gosh Phil, it sure would be nice to have something in writing!" I must also thank Capt Carl Davis for trying to get me started early and his encouragement to do a good job, Lt Col D.J. McBride for her technical guidance and assistance, and Dr Dick Fenno, the finest communicator with whom I have had the pleasure of working, for his assistance in the preparation of this thesis in the appropriate style and format. Finally, I want to thank the 1988 AFIT Cheetahs for their camaraderie and

friendship. Venemus, Vidimus, Cheetahmus, Thesisumus, some more Cheetahmus, and Graduateamus.

Phillip H. Beard

Table of Contents

List of Figures

Abstract

The purpose of this study was to develop an efficient computer-based database management system (DBMS) application to automate manual information processing procedures used by the Air Force Institute of Technology School of Systems and Logistics Graduate Programs Office.

The author was able to create an efficient DBMS application that met the needs of the Graduate Programs Office using the Ashton-Tate dBASE III Plus(TM) DBMS and the Concentric Data Systems R&R Relational Report Writer(R). The application was implemented upon completion.

Volume 1 contains four chapters and an appendix. Chapter I, Introduction, provides background on the AFIT Graduate Program Office and their automated data processing requirements, examines characteristics of good DBMSs, examines DBMS development lifecycle, discusses software selection criteria, and examines four DBMS applications developed in 1987. Chapter II, Methodology, documents the methodology used in developing the DBMS. Chapter III, Findings and Analysis, discusses the programmer's incorporation of good DBMS characteristics presented in Chapter I and discusses whether the author was successful in achieving his goal of solving the specific problem. Chapter IV, Conclusions and Recommendations, describes the impact on Graduate Programs Office operations using the DBMS and

recommends follow-on studies. The appendix, Graduate Programs Office User's Manual, describes DBMS operations.

A DATABASE MANAGEMENT SYSTEM APPLICATION FOR THE GRADUATE PROGRAMS OFFICE OF THE SCHOOL OF SYSTEMS AND LOGISTICS VOLUME 1: DEVELOPMENT AND USER'S MANUAL

I. Introduction

Background

The Air Force Institute of Technology (AFIT) School of Systems and Logistics, located at Wright-Patterson Air Force Base, Ohio, "provides a graduate education in logistics, engineering, and systems management leading to the master of science degree" (9:9). Within the School of Systems and Logistics, the Graduate Programs Office (LSG) manages graduate student affairs.

> The mission of this office is to provide a central point of contact for all graduate students attending the School of Systems and Logistics. It serves as the primary interface with the AFIT Admissions and Registrar Directorate for academic affairs in addition to providing academic support for the six graduate programs in the areas of student operations and records, supply, administration, financial management and the scheduling of classes. Nonacademic and military matters of the graduate students are also provided by the Graduate Programs Office [8:164].

LSG consists of a department head and two academic operations and support personnel. The academic year begins with student orientation at the end of May and ends with graduation at the end of September or December the following

year. The largest class, 177 students, began orientation in May 1988.

LSG performs day-to-day operations and tracks information using manual methods. Extracting data from individual biographical forms and computer printouts requires many man-hours. Requests for information come from within the Air University, the Air Force, the Department of Defense, and from foreign governments whose students attend the School of Systems and Logistics.

"Organizational effectiveness is the degree to which an organization achieves its goals" (7:334). "Efficiency is the amount of resources used to produce a unit of output" (7:335). During the December 1987 AFIT unit effectiveness inspection, Air University inspectors determined LSG effectively performs its mission (10:N-3). LSG acquired a Zenith Z-248 micro-computer in December 1987 and the Ashton-Tate dBASE III Plus™ database management system software in June 1988. The manual data management process LSG uses can be made more efficient by using a computer-based database management system (DBMS) to reduce the time required to produce ad hoc information requests and satisfy management information needs.

Specific Problem

The problem is to develop a micro-computer DBMS application which will efficiently automate the Graduate Programs Office manual data processing procedures.

Graduate Programs Office Requirements

When a student is selected for an AFIT graduate systems or logistics program, the registrar notifies LSG. LSG mails each student an information package that includes a request for biographical data. The biographical sheets have been the foundation for manual data collection and information processing. During student inprocessing, students must provide LSG with an education plan outlining their course of study through graduation. LSG must provide this information to the registrar and each academic program manager for course scheduling, textbook requirements, and instructor scheduling. LSG collects grades at the end of each academic quarter and provides the information to the registrar. LSG must identify and track students who do not maintain academic standards. Students who achieve academic honors are eligible for the Dean's list and honorary society membership. Following last quarter finals, LSG must identify the top ten percent of the class for distinguished graduate consideration. After graduation, LSG compiles historical data on student demographics, curriculum, and academic achievement. Current historical files do not include all information LSG would like to maintain due to the volume of paperwork and limited facilities for data storage.

Literature Review and Software Selection

This literature review identifies characteristics of an efficient DBMS, discusses software selection criteria, and examines prior studies where micro-computer based DBMSs were developed.

Database Management System. A database management system is a method of entering, editing, manipulating, and deleting information contained in one or more databases. The success of a DBMS application depends on the user's computer expertise and knowledge of the application, the user-friendliness of the DBMS, complexity and scope of the DBMS, and the ownership of the DBMS (19:291).

Databases. A database is a collection of data. A telephone book can be considered a database. Each listing has a name of an individual or business, an address, and a telephone number. Each of the pieces of data makes up a record. In the context of computer applications, a collection of data items (or fields) comprise a record. A series of records comprise a file. "A set of files make up a database" (21). Computer databases can take one of three DBMS forms: hierarchical, network, or relational.

A hierarchical DBMS takes on a structure that has been compared to a that of an inverted tree (25:33-34, 1:95-99). A hierarchical DBMS of an organizational supervisor/worker relationship would have the head of the organization at the top level of the database. All subordinate workers would be listed below the organizational leader at the second level.

Employees subordinate to those in the second level would be listed in the third level, and so forth until all employees in the organization were listed. An important feature of hierarchical databases is that a higher-level, or "parent," record can have many subordinate, or "children" records, but a record may have only one parent record (21, 25:33-34). Hierarchical DBMSs are usually very complex in design and make it difficult for a user to insert and delete data.

> Difficulties arise mostly when trying to force a hierarchical structure on data that is not hierarchical; then adding branches (instead of leaves) is difficult. Programming applications for either hierarchical or network databases is tedious because of the record-at-a-time processing required [21].

Additionally, data redundancy may occur (1:106).

When databases can be connected through one or more data items, you have a network DBMS application. Atre describes a hospital DBMS where a database for physicians is related to a patient database. The patient database is related to a surgery database that is related back to the physician database (1:111-117). A network DBMS has two major disadvantages. The first disadvantage is the complexity required to program the relationships between databases. The second disadvantage is programming dependence on the initial DBMS structure. Changing one field in a network DBMS can cause a series of changes in related programs and routines in the DBMS (1:121-122).

A relational DBMS consists of two-dimensional table. Each table is made up of rows and columns. Tables, rows,

and columns are also known by more technical, synonymous terms. A table is sometimes called a relation or file. A row can be a tuple or record. A column is also an attribute or field. Terminology depends on the degree of literature technicality. The most popular terminology is table, row, and column (25:25). Relational systems are generally easier to use because users do not need the in-depth knowledge of the underlying database structure required with hierarchical and network systems (25:29-30, 1:94-95).

Characteristics of a Good Relational DBMS. A good relational DBMS has total data independence. Data independence means that DBMS programming does not rely on the physical structure of its databases. If you have data independence, a database structure can change without requiring reprogramming of the DBMS to incorporate that change (1:16-20, 15:138).

A second characteristic of a good DBMS is controlled data redundancy. Databases should be designed to eliminate all but essential links or key columns. Key columns allow the programmer to link databases through unique data items (20:100).

A third characteristic of a good DBMS is data integrity. Data integrity is the process of ensuring the validity of the data being entered. If a user tries to enter a number in a character field, or tries to enter the

social security number of a person not in the DBMS, the DBMS should not allow the entry (25:15-16).

DBMS Life Cycle. Atre identifies six main phases of a DBMS life cycle: design of the database; physical creation of the database; conversion of the existing data sets and applications to match the newly created database; integration of the converted applications into the new database; the operations phase; and the growth, change, and maintenance phase (1:40-49). Two paths are followed depending on whether the DBMS is being developed from an existing automated information system or a manual system (1:40).

If the DBMS is being developed from a manual system, phases one, two, five, and six apply (1:40). If an automated data set does not exist, there is no need to convert the automated data set to the new DBMS format and phases three and four do not apply. If the DBMS is being developed from an existing automated system and the data files must be converted, all six phases apply (1:40). For the Graduate Programs Office, an existing automated information system does not exist. Phases three and four, data set conversion and integration with the new DBMS, will not apply.

Phase one consists of database design. The two most important steps in this process are the development of data dictionaries and database normalization (1:43, 25:73). If the DBMS designer does not adequately define his or her data

and the inter-relationships between the data, or identify potential flaws in data entry, updates, and queries, the result may be having to restart the DBMS design process.

A data dictionary lists the data items for each database, data item attributes (data type, maximum length of the data item, decimal places for numeric data types), and a description of the data item to include its relationship to other data items.

Normalization is a conceptually complex process of identifying inherent database design problems. The process involves three basic normal forms: first, second, and third normal form. The concept of normalization requires an understanding of two additional terms: functional dependence and primary key.

In a database table, rows are made up of two or more columns. A column of data that makes all columns of information contained in the row it occupies unique within that table is called a primary key. For example, a table of information on a set of employees in a company might have a social security number or unique employee number as a primary key. Two or more people might have the same first name, last name, and middle initial, but the social security number makes the columns of data in that row of a table unique to that person as social security numbers are unique. In this example, any column of data associated with that person (in the same row as the social security number) is said to be functionally dependent on the social security

number column. In the previous example, the name data is functionally dependent on the social security number (25:75-78).

"A relation (table) is in first normal form if it does not contain a repeating group" (25:78). If each column in a row of a table can only contain one data item, the table is in the first normal form (25:78-79).

Second normal form is achieved when a table is in first normal form and each non-key column is dependent on the entire primary key. "Where multiple-column primary keys are necessary or appropriate, all non-key columns should depend on the entire set of columns that form the primary key" (21). Second normal form can be achieved easily by developing a table with a single key column value within that table. A social security number, a unique identification number, or a serial number are examples of key column data types that will provide table normalization in the second normal form (25:79-82).

Third normal form is achieved when a table is in the second normal form and non-key columns within a table are dependent only on the key column in the row in which they appear (21, 25:82-85).

The second phase in a DBMS life cycle is the physical creation of databases. Prototype databases are developed, linked, and tested for satisfactory performance (1:45).

The third DBMS life cycle phase is conversion of the existing data sets and applications to match the newly

created database. This phase is only required when existing automated data and programs that are not in the newly created DBMS format must be manipulated or reprogrammed to fit the new DBMS structure or command language syntax (1:46). This phase is excluded from the LSG DBMS development as there is no existing automated system.

The fourth phase in a DBMS life cycle is the integration of the converted applications and the new applications into the new database. "This phase may be heavily overlapped with phase 3" (1:46). Like phase three, this phase is only required when an existing automated system is incorporated into the DBMS. The user and programmer must ensure that the converted applications will be able to expand and grow as the DBMS expands and grows. If this phase is not properly considered and executed, it could result in a return to the design phase (1:46). This phase is also excluded from the LSG DBMS development for the same reason as given for phase three exclusion.

The fifth phase is the operations phase. "In this phase all applications that are supposed to run using the data base are run full scale" (1:46). Both the user and the programmer must be involved with verification and validation of DBMS functions (1:48) Undetected errors could result in a catastrophic failure (such as loss of data) at a later date.

The sixth phase is the growth, change, and maintenance phase. As users become familiar with the DBMS and begin to

realize its usefulness, more capabilities and functions are desired. The DBMS can be restructured or have utilities and databases added to include new data and information requirements (1:48-49).

Database Administrator. The database administrator "is responsible for supervising both the database and the use of the DBMS." (25:135) The database administrator determines who has physical access to the database, ensures operators receive DBMS training, and ensures the DBMS is kept current and enhancements are programmed when needed (25:134-146). The Graduate Programs Director will serve as the database administrator (28).

Software Selection. "The key to end user database processing is the interface with the DBMS" (19:291). The LSG administrators have limited computer expertise. They work with mainframe computer data input terminals and Zenith Z-248 micro-computers. Micro-computer applications include using spreadsheet templates, but the administrators have no programming expertise. For this reason, the users desire a menu-driven DBMS (28, 12, 16).

LSG has a Zenith Z-248 micro-computer and has recently acquired the Ashton-Tate dBASE III Plus™ DBMS. The School of Systems and Logistics offers a course in DBMSs using dBASE III Plus. Faculty members having dBASE programming expertise can assist the database administrator with phase six DBMS maintenance.

Two AFIT faculty members recognized for their computer expertise recommended dBASE III Plus for an LSG DBMS application. In an informal interview, they cited the flexibility of the dBASE programming language for building menus, the availability of commercially-developed programming documentation, and dBASE's ability to interface with other micro- and mainframe computer software packages as reasons for selecting dBASE III Plus to develop this application (14, 23).

Goley examined dBASE III Plus and two other popular micro-computer DBMS software packages. Because the strength of dBASE's programming language exceeded any limiting factors he considered, Goley recommended dBASE III Plus over two other popular DBMS software packages (13:8-9).

A final consideration in selecting dBASE III Plus is its use of the query language SQL. SQL became the national standard for database query languages in August 1986 (13:49). AFIT is currently in the process of converting its mainframe information processing requirements to the ORACLE DBMS. ORACLE also uses the SQL query format. This will allow the development of programs to transfer data files between the AFIT mainframe computer system and the LSG micro-computer system when ORACLE comes on-line.

Prior Studies. A search of Defense Technical Information Center reports in February 1988 provided four DBMSs developed using dBASE III Plus.

The first report, a March 1987 Naval Postgraduate School thesis, describes the process of developing a relational database application for managing officers in the Korean armed forces. The authors describe the processes for creating a DBMS and describe the specific functions of their DBMS. The application is theoretical as only sample data was used and there was no validation of the DBMS as an effective personnel management tool. The DBMS is menu driven with few output report routines. The outputs that are available use a simple routine for report headers and use the dBASE III Plus "LIST" command for data output. The concept of DBMS design and development is well defined, but it is difficult to determine the usefulness of the application without an end-user evaluation and validation (18).

The second report is also a Naval Postgraduate School thesis. Written in June 1987, this DBMS was created to manipulate data collected from the National Training Center at Fort Irwin, California. The author identified the user's needs and incorporated those needs in his discussion of the DBMS development. The DBMS is menu-driven. The author included a user's manual that documents DBMS capabilities and operation. The DBMS does not contain any security measures to prevent unauthorized access to unclassified but sensitive information. For that reason, the author only provides computer screen displays for data output. The DBMS has no built-in hard-copy output capability. The author did

not validate all DBMS functions and includes a disclaimer at the beginning of the thesis (3). The DBMS application has not been implemented in the field. A data analysis system using a mainframe computer is still used to assess unit performance at the National Training Center (17).

The third report is a DBMS developed to manage Department of the Army underground storage tank data. This report, created by the U.S. Army Construction Engineering Research Laboratory (USA-CERL) documents the development and structure of DBMS database tables. It does not detail general DBMS characteristics nor outline the steps required to build a DBMS. The report does not discuss DBMS validation or the success of the DBMS in managing underground storage tank data (11). Environmental engineers at two Army major command headquarters were trained on DBMS use (22). As of the end of August 1988, the DBMS had not been implemented and further DBMS modifications were expected by major command personnel (24). A potential major command user and the development team's supervisor were not aware of the DBMS implementation status (22, 24). This researcher was asked by the development team supervisor to relay any negative user comments collected during the user interview (22). The researcher asked the user to contact the development team's supervisor to provide feedback on the storage tank DBMS (24). The researcher assumes the DBMS developers and users did not have a close working relationship during the DBMS development process. This lack

of communication may be a contributing factor in the failure to implement this DBMS.

The fourth report is a September 1987 AFIT thesis that created a DBMS application to aid U.S. Navy supply personnel in producing fleet ballistic missile tender daily high-priority requisition reports. The author interviewed Navy supply personnel to determine DBMS data requirements. She then developed and tested the application by having experienced Navy supply officers use the DBMS. The validity of the DBMS application was determined by AFIT faculty members who were identified as having DBMS/management information system expertise. There was no actual testing and validation in the operational environment by potential users. The majority of the thesis content deals with program documentation and a quasi-user's manual description of the DBMS capabilities and operation (27). The Navy is currently converting shipboard computer systems and has not implemented this DBMS. The initial program does meet the user's needs for a stand-alone system, but the Navy will look at integrating the stand-alone DBMS with mainframe systems to better access and reduce data input requirements (2).

Scope of the Thesis

This thesis will result in a micro-computer based DBMS application with an accompanying user's manual and technical reference manual. This thesis will not deal with specific

programming techniques or provide a detailed background of the dBASE III Plus structured programming language except where programming techniques or specific dBASE commands must be explained for future additions and maintenance.

Only data required by the Graduate Programs Office will be considered as input to the database. Known data output requirements will be programmed as menu options and specific report formats will be developed. LSG provides information for ad hoc requests as they are received. To the extent practical, the programmer will provide a framework for ad hoc data queries. It will be incumbent on the users to develop a working knowledge of dBASE III Plus query commands to take full advantage of this DBMS.

Organization of the Thesis

This thesis contains four chapters and two appendices. Chapter I is the introduction containing LSG background information, a statement of the specific problem, rationale for software selection, and a literature review of applicable DBMS characteristics, design process, and some existing DBASE III Plus DBMSs.

Chapter II details the thesis methodology used to solve the problem. This chapter describes data collection, the database design process, prototype development, field tests, modifications, DBMS validation, and implementation.

Chapter III describes the design and development outcome and provides an analysis of the DBMS with respect to

the original goal of creating an efficient DBMS for the Graduate Programs Office.

Chapter IV describes the results of the DBMS application and its strengths and weaknesses. This chapter also provides recommendations for follow-on studies.

The Appendix is the Graduate Programs Office DBMS User's Manual. This users manual documents DBMS operations and provides examples of ad hoc query commands. Some potential troubleshooting procedures are provided in case the DBMS structure is changed through ad hoc operations.

II. Methodology

Introduction

The process of developing a DBMS for the Graduate Programs Office consisted of data collection, database design, prototype design and testing by the developer, user testing, DBMS validation by AFIT Faculty experts, and DBMS implementation.

Data Collection

The first step in data collection began with an interview of the Graduate Programs Director to determine what information he would most often require as DBMS output or reports (28). Initially, the Dean of the School of Systems and Logistics and department chairmen were considered as part of the interview process, but in limiting the scope of the DBMS environment and development effort, only the Graduate Programs Director was interviewed. Information managed by the Graduate Programs Director fell into three categories: demographic, academic, and course information.

Student demographics information included data such as name, rank, rated/non-rated status, married/single status, military seniority lists for promotion and student leadership positions, educational background, military background, and next assignment information. Students

provided demographics information on a form sent to them by LSG when information packages were mailed in the spring prior to May in-processing. The front and reverse of the form are shown in Figures 1 and 2.

Course information included edplans for each of the ten degree programs in the School of Systems and Logistics and the courses that would be offered in a given academic quarter. For each course offered in a quarter, LSG had to provide course managers with the number of students in each course and the names of those students. This information was used to determine the number of sections and instructors for each course.

Academic information requirements included student education plans (edplans) and academic performance. Each student received a generic edplan for their program of study and selected elective courses according to their degree program. Students return completed edplans to LSG sometime prior to the beginning of the summer short term. If a student changed his or her edplan during their course of study, the student had to provide the changes to LSG. Figure 3 depicts the AFIT form used for curriculum changes. At the end of each academic quarter, instructors forwarded course grades to LSG. LSG needed the capability of identifying students with cumulative GPAs below 3.0 or course grades of "U", "D", or "F" for academic probation. Students with cumulative GPAs above 3.75 received Dean's

DATA FORM

PRIVACY ACT STATEMENT
AUTHORITY: 44 USC 3103, E.O. 9397
PRINCIPLE PURPOSE: Used for locator, emergency notification, to build class rosters, and develop statistical reports and analyses. A ready reference for on-going courses and development of new courses. Description of individual's present position, education, experience, student evaluation, and final grade record.
ROUTINE USES: As a locator, emergency notification, to build rosters, develop statistical reports, and for evaluation. May be used by various base agencies in the performance of their required duties.
DISCLOSURE IS VOLUNTARY: Failure to provide the information requested could hinder fast and efficient action in emergency situations or prevent development of statistical analyses for future requirements. (SSAN is necessary to make positive identification of individual and records.)

NAME:_____NICKNAME:_____
Last Name, First Name, Middle Initial

GRADE/RANK:_____SSAN:_____SEX:_____

LAST MAJCOM:_____BRANCH OF SERVICE:_____AFSC:_____

PROGRAM YOU WILL BE STUDYING HERE:_____

DATE OF BIRTH:_____ DATE OF RANK:_____
Year-Month-Day Year-Month-Day

YEARS OF SERVICE:_____ SERVICE DATE:_____
Year-Month-Day

ETHNIC GROUP: White_____ Black_____ Hispanic_____

Asian/Pacific Island_____ American/Alaskan Indian_____

AERO RATING:_____ LAST PLANE FLOWN IF RATED:_____

MARRIED/SINGLE:____ SPOUSE'S NAME:_____NICKNAME:_____

WILL SPOUSE ACCOMPANY YOU TO AFIT:_____ IS SPOUSE MILITARY:_____

NO. CHILDREN:_____ IF SINGLE, ANY DEPENDENT CHILDREN:_____

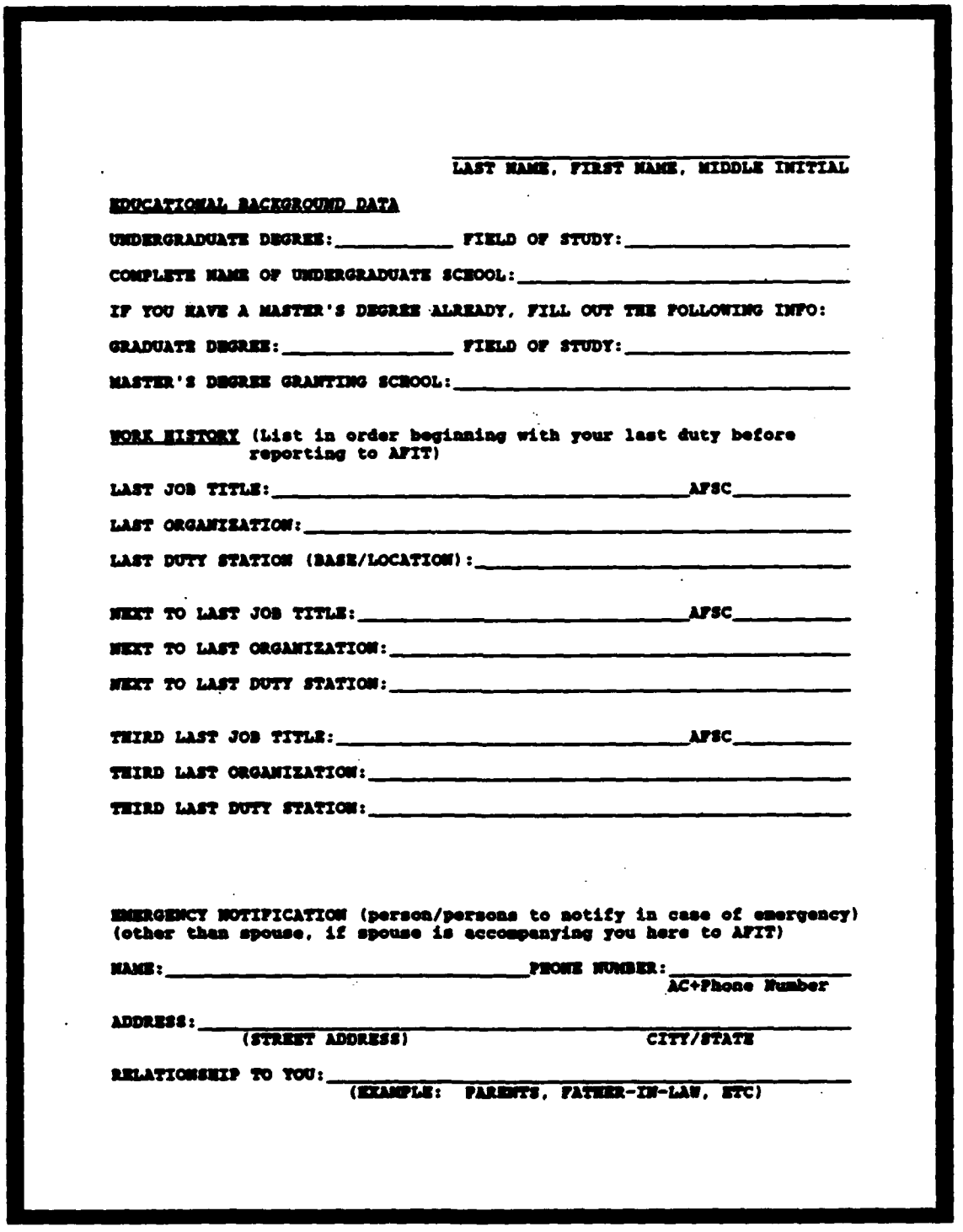
WILL DEPENDENT CHILDREN ACCOMPANY YOU HERE TO AFIT_____

ADDRESS (IF KNOWN) _____
(STREET ADDRESS YOU WILL BE LIVING AT HERE)

_____ _____
(CITY AND ZIP CODE) HOME PHONE NUMBER

----------PLEASE FILL OUT THE REVERSE OF THIS FORM ALSO----------

Figure 1. Student Demographics Data Form (Front)



LAST NAME, FIRST NAME, MIDDLE INITIAL

EDUCATIONAL BACKGROUND DATA

UNDERGRADUATE DEGREE:_____ FIELD OF STUDY:_____

COMPLETE NAME OF UNDERGRADUATE SCHOOL:_____

IF YOU HAVE A MASTER'S DEGREE ALREADY, FILL OUT THE FOLLOWING INFO:

GRADUATE DEGREE:_____ FIELD OF STUDY:_____

MASTER'S DEGREE GRANTING SCHOOL:_____

WORK HISTORY (List in order beginning with your last duty before reporting to AFIT)

LAST JOB TITLE:_____AFSC_____

LAST ORGANIZATION:_____

LAST DUTY STATION (BASE/LOCATION):_____

NEXT TO LAST JOB TITLE:_____AFSC_____

NEXT TO LAST ORGANIZATION:_____

NEXT TO LAST DUTY STATION:_____

THIRD LAST JOB TITLE:_____AFSC_____

THIRD LAST ORGANIZATION:_____

THIRD LAST DUTY STATION:_____

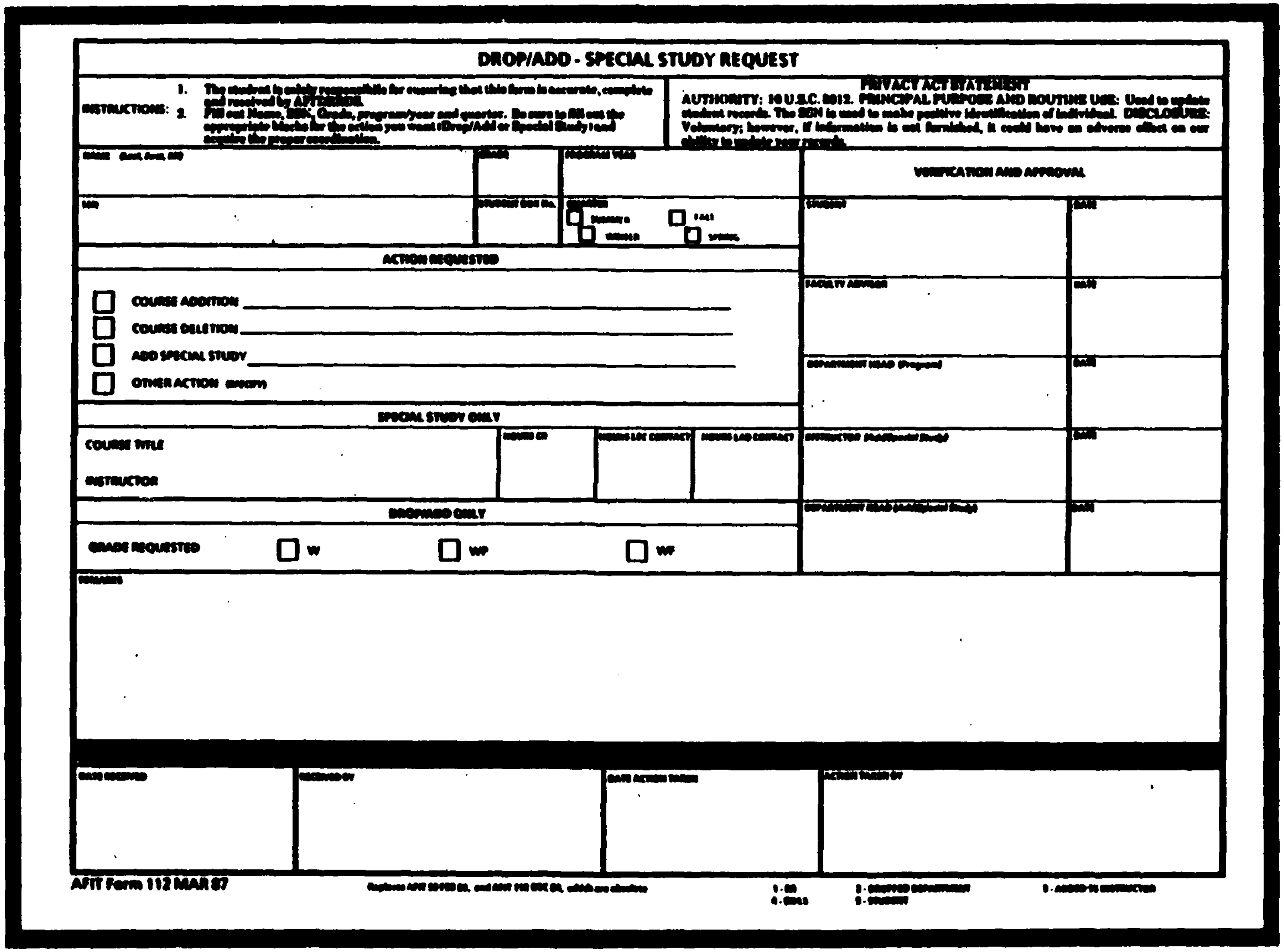
EMERGENCY NOTIFICATION (person/persons to notify in case of emergency) (other than spouse, if spouse is accompanying you here to AFIT)

NAME:_____PHONE NUMBER:_____
AC+Phone Number

ADDRESS:_____
(STREET ADDRESS) CITY/STATE

RELATIONSHIP TO YOU:_____
(EXAMPLE: PARENTS, FATHER-IN-LAW, ETC)

Figure 2. Student Demographics Data Form (Reverse)



DROP/ADD - SPECIAL STUDY REQUEST

INSTRUCTIONS: 1. The student is solely responsible for ensuring that this form is accurate, complete and received by AFIT/[illegible].
2. Fill out Name, SSN, Grade, program/year and quarter. Be sure to fill out the appropriate blocks for the action you want (Drop/Add or Special Study) and acquire the proper coordination.

PRIVACY ACT STATEMENT

AUTHORITY: 10 U.S.C. 8012. PRINCIPAL PURPOSE AND ROUTINE USE: Used to update student records. The SSN is used to make positive identification of individual. DISCLOSURE: Voluntary; however, if information is not furnished, it could have an adverse effect on our ability to update your records.

VERIFICATION AND APPROVAL

ACTION REQUESTED

COURSE ADDITION

COURSE DELETION

ADD SPECIAL STUDY

OTHER ACTION

SPECIAL STUDY ONLY

COURSE TITLE

INSTRUCTOR

DROP/ADD ONLY

GRADE REQUESTED W WP WF

AFIT Form 112 MAR 87

Figure 3. Drop/Add - Special Study Request (AFIT Form 112)

List consideration. Additionally, the Graduate Programs Director wanted the capability to list students by cumulative GPA at the end of each academic quarter. This information would aid in identifying Distinguished Graduate candidates for graduation in September and December.

Throughout the DBMS design process, the programmer worked closely with the Graduate Programs Director to ensure LSG needs were being met. As additional reports and DBMS capabilities were identified, the Graduate Programs director was consulted to determine if he wanted to incorporate those ideas into the DBMS.

Database Design

The creation of database tables followed the sequence used by LSG in processing a student class. The initial step in the data collection process required obtaining student demographics information for the incoming class. Degree program managers would provide LSG with updated copies of degree edplans prior to student in-processing. LSG would then distribute edplans for students to complete. Student edplans were forwarded to the registrar and became the basis for academic transcripts. Instructors provided course grades at the end of each academic quarter and LSG tracked student academic progress through graduation. Finally, LSG stored graduate records for future analysis.

Database tables were created using the dBASE III Plus Assist Program, a built-in application. The programmer used

the Assist Program to create database dictionaries, organize the data based on one or more key columns, and input data. Database dictionaries, screen format files, and associated database index files are documented in Volume 2: Graduate Programs Office DBMS Technical Reference Manual.

The Student Table was the first table created. All data was normalized in the third normal form. The programmer created custom data entry screens from the Assist Program replicating the format of the demographic data entry form. Maintaining a screen data entry format that matched the input source should facilitate user data entry. dBASE Edit instructions were provided to the users by programming them on custom built screens. The programmer input demographics data for Class 88S/D and available information for Class 89S/D. This data entry process required entering over 50 data columns for each of the 336 students in the 1988 and 1989 classes.

Throughout the database development process, the programmer accomplished all data entry. By accomplishing data entry himself, the programmer was better able to determine potential problems and solutions in the menu data entry process that had yet to be developed. The programmer was better able to develop DBMS programs to limit user data entry to the proper data type and format.

Next, the ten generic degree edplans were created for Class 88S/D and data entered by the programmer. Two related tables were created during the process of

normalizing edplan data. A Course Table was created to store all data related to the course number: course name and a logical field identifying the course as a graduate level course. This table is normalized in the third normal form. A Terms Table stores the name of the quarter designator against its DBMS designator and was also normalized in the third normal form. Each edplan required three data column entries for at least 25 rows of data per table.

The first "quarter" in the DBMS is the Entering Credit quarter, designated Quarter 0. The last quarter is Quarter 8, the second Fall Quarter for Graduate Information Resource Management (GIR) degree students only. To the users, the designations Quarter 0 and Quarters 6 through 8 do not exist. The reason for using these designations was to facilitate program looping based on numeric ranges rather than a use cumbersome character string quarter identification code. The Terms Table is for report generation only and is not accessed during data entry. The Terms Table relates to generic edplan tables through the "Quarter" key column in edplan tables. Generic edplans relate to the Student Table based on the "Option" key column in each row in the Student Table.

The next step in the process was to create specific student edplans for Class 88S/D; the Academic Table contains all student edplans. A program identified each Class 88S/D student in the Student Table, matched their degree program (Option column) in the Student Table with

the corresponding generic edplan table, and copied all records in the generic edplan table to the Academic Table. Class 89S/D student records were filtered out of the selection process by using the dBASE "SET FILTER TO" command. When all student edplans were created for Class 88S/D, the generic edplans were updated with Class 89S/D edplan data using the Assist Program. The program filter was changed to prevent Class 88S/D students from being re-selected for edplan creation. Class 89S/D student edplans were appended to the end of the Academic Table. The Academic Table contains over 9600 rows containing ten data columns. The Z-248 computer took over 40 minutes to create this table. When the Academic Table was built, the programmer updated each student's edplan for electives and input letter grades for Class 88S/D students. This process required manually editing over 1000 records in the Academic Table to enter electives. Grades were only available through the 1988 Spring Quarter. The programmer entered an average of four grades per student per academic quarter. There were a total of 159 students in Class 88S/D and five quarters of grade inputs. The key column relating the Academic Table to the Student Table was the "SSAN" key column. The Academic Table was normalized in the third normal form.

The next table created was the Grades Table. This table stores each student's current and cumulative hours, academic credit points, and grade point averages (GPAs) by

academic quarter. This data is generated by a DBMS program developed by the programmer. The Grades Table was normalized in the third normal form. The Grades Table relates to the Academic Table through the "SSAN" key column in the Academic Table.

LSG generates a quarterly grade statistics summary. To automate this function, the Totstats, GCAStats, GSMStats, GCMStats, GEMStats, GIRStats, GLM1Stat, and GLM2Stat Tables were created. These tables store DBMS program results of dBASE "COUNT" and "AVERAGE" commands used on columns in the Grades Table. Each table contains at least 42 columns and is normalized in the third normal form. Some tables contain a combination of two or more sets of independent data and therefore vary in length. The reason separate Tables were not created is due to a limitation in the R&R Relational Report Writer(R) report generator (5). The report generator can only relate a maximum of ten tables. If each program option were structured as a statistical table, the report generator would have been unable to link columns containing statistical totals and counts contained in the Totstats Table. See Volume 2: Graduate Programs Office Technical Reference Manual for a description of data dictionaries. All tables relate to the Grades Table through the key column "Field."

Another LSG administrative function, identified during the data collection process, is to determine course offerings by academic quarter. The Loadings Table stores

course numbers and number of students for that course for a user specified academic quarter. This data is created by a DBMS program that scans the Academic Table for courses in a user specified quarter. The Loadings Table is normalized in the third normal form and is not related to any other table in the DBMS.

The PASCode Table resulted from normalizing graduating students' next assignment codes stored in the Student Table. This table stores a two-digit personnel assignment selection (PAS) code, the major command/agency acronym, and the name of the major command/agency. The programmer entered data using the Assist Program. The table contains over 50 rows of three data columns. All data is normalized in the third normal form and the key column link to the Student Table is the "PASCode" column.

The last two tables are the RRUNIN and RRUNOUT Tables. These tables are pre-programmed database modules provided with R&R Relational Report Writer, a dBASE report generator (5). The report generator uses the RRUNOUT Table to store report errors and status. The table is not related to other tables and is normalized in the third normal form. The RRUNIN Table stores report data used to generate reports using dBASE III Plus programming code. This table is normalized in the third normal form and does not relate to other DBMS tables. The programmer entered RRUNIN data using the Assist Program following development of the 61 DBMS reports.

Prototype Design

Prototype design began with creation of a menu system to allow users to perform frequent tasks. These tasks included adding data to the DBMS, editing data in the DBMS, deleting DBMS data, calculating grades, generating reports, backing up and restoring DBMS data and programs, exiting the system, and performing ad hoc DBMS operations. Each task was broken into sub-tasks and sub-menus were created to provide user access. All menus are based on program files written by Simpson (26:932-936). Initial modification to these files were required for text alignment. When testing menu routines, problems were encountered when returning to a menu from the next lower menu. Simpson's menu program was not designed to handle multiple sub-menus and further modification to Simpson's program code was required. Menus, displays, and program operation are explained in the Appendix, Graduate Programs Office DBMS User's Manual.

Each menu option was handled as a separate module to simplify the programming task. Options on the main menu screen are logically grouped by task type as procedure files within each main program. dBASE III Plus allows only ten program files to be open at once. If program files are chained together, the eleventh program called-up in a chain will cause a dBASE error for too many open files. dBASE closes a procedure file prior to using a follow-on procedure file avoiding the too many open files error.

DBMS programs are written using dBASE III Plus structured programming commands (26). The report generation program uses R&R Relational Report Writer execution (.EXE) and configuration (.CNF) files to access pre-programmed reports.

R&R Relational Report Writer uses dBASE generated tables and index files to create reports. The programmer uses R&R to create report formats. The report development process was made easier by creating an initial report and then modifying it for different academic quarters or student demographics data requirements and saving the new report under a different name. Duplicating academic reports for different quarters required changing report header information and data queries. Student reports required changing report header information and data queries. Reports can be displayed or printed from R&R, from dBASE programs, from the dBASE command prompt, or from the disk operating system.

Problems were encountered in integrating pre-programmed DBMS reports with dBASE programs. To provide the flexibility necessary when printing data for multiple student classes, DBMS programs allow the user to specify report parameters that are passed from a DBMS program to the report generator. Report parameters were programmed as memory variables and concatenated as string text and stored in the RRUNIN Table. R&R Relational Report Writer did not provide clear examples for the level of programming

complexity in this DBMS. The programmer had to call Concentric Data Systems technical consultants for guidance while creating initial dBASE programs.

As each report format was completed, the report was generated both as a screen display and as printed output. Reports matched existing LSG reports in format and structure as much as possible. The Graduate Programs Director and LSG administrative personnel were given report output and ask to review the output for format and data content. Previous reports generated by the AFIT Registrar did not allow report customization. Recommendations for DBMS report enhancements and additional data items were common. Refer to Volume 2: Graduate Programs Office DBMS Technical Reference Manual for report specifications.

To reduce programming complexity and facilitate future DBMS maintenance, the Academic and Grades Tables contain redundant data items contained in the Student Table. The Academic Table contains columns for last name, first name, middle initial, and class in addition to the key column "SSAN." This data redundancy allows data to be organized alphabetically and by course and quarter. When two student classes overlap, it is necessary to identify and organize academic records by class year.

Initial database design resulted in minimized data redundancy. When dBASE relations were specified and data editing attempted, the programmer determined the process was too slow and cumbersome to be effective. By adding name and

class redundant columns, editing times were reduced from an hour to under ten minutes.

If a DBMS contains data redundancy, the programmer must ensure common data columns are updated in all tables if a data item changes. Data items common to more than one table are name, SSAN, class, and option. A special edit module allows users to change these columns in each table. In the case of "SSAN," a key column in the Student, Academic, and Grades Tables, the tables are edited and the index files are reindexed to update any changes. Refer to Volume 2: Graduate Programs Office DBMS Technical Manual for specific EDIT.PRG programming documentation.

Throughout the programming process, the programmer tried to eliminate the potential for user input errors. Where users input numbers corresponding to an input option, a range of values was specified to preclude the user trying to input a value outside of the specified range. In the case of course designations, a dBASE Picture Function was used to ensure the first four characters input would only be upper case letters and the last three characters would only be numbers. Where input required a social security number, a Picture Function pre-positioned dashes ("-") so the user only had to input numbers. International students are assigned numbers equivalent in length to social security numbers and beginning with "999-99-", the last four digits varying with each student. In the case of screen displays, key columns and redundant data columns were formatted to

prevent the user from accidentally changing the data. When tables were edited, all related index files were opened to ensure data was updated and organized properly. For any DBMS operation involving individual student, academic or grade data, the user must specify the student's social security number (SSAN). All programs check to ensure the SSAN is a valid student SSAN before the user can perform any operation.

Prototype Testing

As modules were completed, the programmer tested them using actual database data. Errors in command syntax and typing were minor and corrected as they were discovered. When errors were corrected, the programmer retested all programs within that main program file to ensure proper program execution. Prototype development and testing was done on an NEC Multispeed HD laptop computer. After prototype testing, all DBMS files were transferred to 5.25 inch diskette and loaded on the LSG Zenith Z-248 microcomputer. All reports were printed on the LSG printer and given to LSG administrative personnel for verification of the data entered by the programmer during database development.

Field Test

The programmer and the Graduate Programs Director jointly verified the DBMS operation using the LSG microcomputer. The test consisted of adding faculty members

as students using class designations of 90S or 90D. Fictitious social security numbers were used as they were easy to remember. Four student records were created using only the name, SSAN, Class, and Option columns. All other columns were left blank. Next, the generic edplans were edited to test the edit module. Data was re-edited back to original values to preclude errors after actual DBMS implementation. Student edplans were created for the faculty "student records" to validate that procedure. Two edplans were then edited to add specific courses in place of elective course designations. Summer Short Term course grades were input to validate the course grade edit module. The programmer's grades were edited to validate the individual grade edit module. Grades were changed back to their original values to ensure DBMS accuracy after implementation. The quarter grade calculation module was tested to validate its operation. The Graduate Programs Director validated report generation programs by selecting report options and displaying or printing them. The programmer had hard copy output for all reports to validate the creation of a desired report.

The Graduate Programs Director was satisfied the DBMS would meet LSG needs and requested minor changes to reports and user interface screens. Recommended changes were incorporated and tested by the programmer and Graduate Programs Director.

Validation by Faculty DBMS Expert

A faculty member who teaches management information system and DBMS structures courses in the School of Systems and Logistics was given a demonstration of the DBMS and asked to critique the application (21). Recommendations for DBMS improvement and design concerns included regrouping some tasks using a more logical arrangement, providing program exit options, and verifying GPA calculations.

The programmer had originally designed historical data file creation and DBMS files maintenance tasks in separate menus. One critique was to reorganize these tasks under a sub-menu of the main menu. Following the initial expert validation, the recommendation was incorporated.

A second critique concerned data integrity. Some columns in the Student Table were not checked for data entry standardization. This lack of standardization could cause incomplete data queries if these columns were used in the query definition. The programmer and the database administrator had discussed this issue previously and they decided that administrative procedures could be established to ensure data standardization.

A third critique was that logical fields in the DBMS defaulted to "T" and "F" for true and false when displayed for user prompts. Users might not understand the "T/F" convention and a "Y" and "N" convention for yes and no was recommended. The convention was changed for the validation review.

The procedure that allowed edplan editing contained a procedure to add courses to the Course Table if it was not already in the Course Table. This routine did not allow the user the option of exiting the Course Table add procedure if the user accidentally typed an incorrect course designator. This was a significant oversight in the initial program design and was corrected following the initial expert validation.

The initial DBMS program did not always notify the user of program execution or report generation while tables were being scanned. There was some concern that the user might not be aware a program was running and attempt to re-enter data prompts displayed on the screen. This potential problem was solved by adding routines to clear the screen and display advisories during program execution.

Since this DBMS will calculate GPAs and be used to determine potential distinguished graduate students, the instructor was concerned that the DBMS use the same equation for GPA calculation as used by the registrar. The programmer compared DBMS calculated GPAs with registrar generated GPA rosters and determined the DBMS process to be the same. The registrar was contacted to confirm the GPA calculation processes were the same and they were (6, 4).

Database Implementation

Following expert validation, the implementation process began. LSG administrators were given a training session

demonstrating the final DBMS. Training consisted of hands-on orientation with the programmer answering DBMS operations questions. The user's guide was briefed and examples of DBMS operations demonstrated. The technical reference manual and possible enhancements such as the use of a compiler and potential interface with the AFIT Oracle DBMS under development on mainframe computers were discussed with the database administrator.

The status of data input and update requirements were briefed to the database administrator. Data was entered based on information available at the end of the 1988 spring quarter. To update the DBMS, grades and student edplan changes since the spring quarter were remaining data entries.

The DBMS development process took place over a three month period. The programmer averaged approximately 30 hours per week building tables, entering data, and developing DBMS programs and reports.

This chapter described the methodology used to develop this DBMS application. The next chapter discusses findings during DBMS design and analyzes good DBMS characteristics with respect to the original goal of developing an efficient, automated DBMS application for the Graduate Programs Office.

III. Findings and Analysis

Introduction

This chapter describes the DBMS development process and analyzes the success the programmer achieved in incorporating characteristics of good databases in the Graduate Programs Office DBMS. Programmer findings and a discussion of the Graduate Programs Office DBMS lifecycle are documented in addition to an assessment of the software used in DBMS development. The chapter ends with an analysis of the DBMS with respect to the original goal of solving the specific problem as stated in Chapter I.

Incorporation of Characteristics of Good Databases

Data Independence. As stated in Chapter I, data independence means that DBMS programming does not rely on physical database structures. The programmer was not able to incorporate data independence in DBMS programs available to the user through the DBMS menu system. To provide a menu-driven system for users with limited database knowledge and computer expertise, many programs relied on database structures in their execution. DBMS reports using R&R Relational Report Writer are directly dependent on database structures (5). Any modifications to database structures will require changes to DBMS programs. Before the database administrator approves any changes to the DBMS, all program

listings should be reviewed to determine the extent of additional programming required.

For ad hoc database operations that will not access DBMS programs, excluding the use of pre-programmed R&R Relational Report Writer reports, listings and queries can be performed without concern for database structure. By entering individual dBASE commands from the dot prompt or using the Assist Program, the user can perform DBMS operations with complete data independence. Because the user accesses DBMS tables without regard to their structure, data independence for ad hoc operations has been achieved by virtue of the data manipulation language that is part of the pre-programmed dBASE III Plus DBMS.

Controlled Data Redundancy. Controlled data redundancy was defined in Chapter I as the elimination of all but essential links or key columns in DBMS tables. The original DBMS design minimized data redundancy. The only redundant data in DBMS tables were key columns, such as SSAN, necessary to link and relate tables. The programmer determined that initial edit and query operations were degraded in terms of the time required to perform those operations. By redesigning data dictionaries and including two or three redundant Student Table columns in the Academic and Grade Tables, edit and query functions were reduced to a fifth or sixth of their original operation time. Faster program execution was achieved because the program only accesses one table that has been organized for

the particular operation being performed. The initial program related and searched three tables while performing edit and query operations.

The goal of controlling data redundancy must be balanced with other considerations such as program operating speed and complexity of the programming task. The Graduate Programs Office does not have a full-time computer programmer who can work with complex programming concepts. To keep database relations and programming techniques as simple as possible and use tables singly, rather than creating complex relational chains, a certain amount of data redundancy was appropriate for this application.

Data Integrity. The definition of data integrity given in Chapter I was the process of ensuring data validity with respect to data type, range of values, or standardization of data within table columns or DBMS user inputs. Where tables contain redundant data and that data must be edited or changed, the programmer has ensured all duplicate columns will be edited or changed simultaneously. Where data entry requires a specific type of data, alphabetic, logical, numeric, or a combination of alphabetic and numeric characters, the programmer has tried to ensure the user can only input that data type or format.

The most critical entries users will make are student course grades. Many reports and rosters are dependent on student grade point averages calculated from grade inputs. The programmer has ensured only legitimate grades can be

entered by the user through the menu system.
For ad hoc DBMS operations, there is no way to check for data integrity unless the user selects a screen input format that contains picture functions for certain data columns. The Ashton-Tate pre-programmed dBASE data manipulation language does not allow a programmer or user to specify data integrity checking procedures such as picture functions outside of user or programmer developed screen input formats.

DBMS Lifecycle

As described in Chapter I, phases one, two, five, and six apply to the Graduate Programs Office DBMS. The DBMS is in phase five, the operations phase of its lifecycle. The only problem encountered in this phase was updating the DBMS with student and academic data not updated since initial data entry by the programmer. This problem was administrative in nature and did not affect DBMS development or implementation.

In phase one, database design, the development of data dictionaries was routine. Because of the necessity for data redundancy, additional edit programming was required to ensure all redundant data was updated simultaneously in each table.

Phase two, database creation, was accomplished on a laptop computer. The Graduate Programs Director used his Zenith Z-248 computer extensively precluding the programmer

from having free and unlimited access to the Z-248 for database design and program development. Problems were encountered during the transfer of database, index, and program files from the programmer's laptop computer to the Graduate Programs Director's Z-248. In transferring programs and testing modules on the Z-248, index files were not always current with their associated tables and query operations and report generation could not be successfully performed. Damaged index files were reindexed using the latest database data. By this phase in the development process, tables were current and most DBMS modules were programmed. To preclude additional testing problems, all further DBMS programming and testing was done exclusively on the Z-248.

During phase two, the data entry process was time consuming and lengthy. However, the programmer developed an insight to potential user data entry problems that might not have otherwise been recognized. Also, by entering the data himself, the programmer avoided delays in DBMS programming and testing because of the lack of available persons for data entry.

DBMS Software Selection

dBASE III Plus's structured programming language and commercially available documentation were major factors in developing this DBMS. The programmer had extensive experience in computer programming techniques with BASIC,

FORTRAN, and COBOL languages. This was the programmer's first dBASE programming effort. dBASE was easy to use and followed structure and syntax rules found with languages previously used by the programmer. One commercially available text on dBASE programming provided initial examples and programming tips (26). Other students in the School of Systems and Logistics were also developing dBASE III Plus applications as part of their thesis effort. The synergy created by discussing programming and database issues and techniques was invaluable in avoiding problems encountered by fellow students.

The flexibility dBASE provided in creating menus and dBASE's use of SQL will meet user requirements for a menu-driven DBMS and allow more experienced database users to perform ad hoc operations without the burden of having to access menus. A dBASE III Plus weakness was its inability to easily program reports, particularly complex reports such as those required with this DBMS application. This weakness was overcome through the use of R&R Relational Report Writer.

R&R Relational Report Writer was invaluable in reducing report development time and creating reports to match existing products used by LSG. The programmer had to learn to use the report generator prior to report design in phase two. The report generator came with a well-written, well-documented tutorial module. The documentation on report generation integration with dBASE structured

programming was not as well documented. The programmer had to contact the Concentric Data Systems technical support division to clarify integration concepts and parameter-passing program coding essential to the DBMS operation. The School of Systems and Logistics acquired this package just prior to the database development phase.

Solving the Specific Problem

As stated in Chapter I, the original goal was to develop a micro-computer DBMS application which would efficiently automate the Graduate Programs Office manual data processing procedures. The Graduate Programs Director stated that this DBMS has eliminated the need to manually search computer listings and gives him the capability to access information both efficiently and effectively (28). LSG administrators stated that tasks such as developing class photo books, computing grade statistics, and determining course loads and developing class rosters often took at least a week of intensive work--the DBMS accomplishes these tasks in less than an hour (12, 16).

During the expert review and validation by the faculty instructor, the original academic problem statement was explained and the instructor was asked if she believed the application met the goal of efficiently automating the Graduate Programs Office manual data management procedures. It was her opinion that the goal of solving the specific problem has been achieved (21).

The Graduate Programs Office DBMS application has efficiently and effectively automated manual data processing procedures. To the degree practical in this application, and given the programmer's limited experience with database design, the DBMS application incorporates characteristics of good DBMSs and meets the user's requirements for automated data processing and data management. Chapter IV summarizes the results of the application, identifies DBMS strengths and weaknesses, and makes recommendations to improve the DBMS application and for follow-on studies.

IV. Conclusions and Recommendations

Introduction

This chapter summarizes the results of the DBMS application and identifies strengths and weaknesses. The chapter concludes with recommendations for follow-on studies.

Results of the Application

The Graduate Programs Office DBMS has simplified LSG information management. Tasks associated with quarterly reports and grade analysis have been efficiently programmed in the DBMS allowing administrators to focus on problem solving, not problem identification. Students who require faculty attention as a result of deficient academic performance can be identified within minutes of end-of-quarter grade entry.

Prior to DBMS development, the Graduate Programs Director relied on AFIT Registrar computer products for grade information. These products often took up to three weeks following the end of an academic quarter to be processed and returned to LSG (28). Now, the Graduate Programs Director has access to grade information as soon as it is entered in the DBMS.

Ad hoc data queries can be accomplished in less than fifteen minutes with only minor dBASE command familiarity. Previous ad hoc requests required the Graduate Programs

Director to search multiple computer listings or seek information directly from students (28).

As users become more familiar with dBASE commands and DBMS capabilities, they will be able to provide instructors with student demographic information for a particular course, giving the instructor a better understanding of the students in their class. Academic advisors could be provided with statistics on student advisee performance from quarter to quarter to better assist students who might need attention..

This DBMS has made the Graduate Programs Office more efficient in managing student and academic data. The DBMS has reduced the time required to process information requests and allows the Graduate Programs Director immediate access to data for decision-making. By expanding this DBMS application as additional capabilities are realized, the Graduate Programs Office can continue to recognize efficiencies in day-to-day operations. Given the existing DBMS framework, additional capabilities can be added within hours using existing tables and programs.

DBMS Strengths and Weaknesses

One of the strengths of the DBMS is its ability to organize data and allow the user to retrieve that data in whatever form the user requires. By using the ad hoc feature of the DBMS, user operations are limited only by the user's knowledge of dBASE features and commands. Help is

available through the Assist Program's help utility. If a user enters a command incorrectly during an ad hoc operation, the Assist Program will allow the user to access help information on any dBASE command.

Another DBMS strength is the data integrity verification process during menu operations. Unless a user intentionally enters incorrect data, DBMS programs will ensure proper format and data type before storing data in a table.

By using R&R Relational Report Writer to modify existing reports, any report can be changed or customized to meet the user's needs. As long as the user does not save report modifications under the initial DBMS report name, report format changes will not affect future DBMS menu report generation.

Instructors in the School of Systems and Logistics are using the report generator to research student demographics information that will be used to better assist students in the preparation of their edplans and identification of possible thesis topics. Instructors can determine a student's previous background and duty experience and match that experience with their service's needs for research. The preparation of this report took less than an hour. The majority of that hour was spent developing the report format desired by the instructor. Prior to DBMS implementation, this report would have required at least a day to prepare

and could not have been re-created for future classes without manually searching data forms and rosters.

DBMS weaknesses include the inability to ensure data integrity during ad hoc operations. The programmer has stressed the need to use the menu system for data entry to ensure data integrity to the maximum extent possible.

It is incumbent on the user to fully understand the DBMS structure and interrelationships between tables and index files when performing ad hoc data editing. If the user does not ensure all index files associated with a table are specified when data is added, deleted, or changed, the DBMS will not function properly. The programmer strongly recommends that users perform all data addition, deletion, or editing from the menu system to preclude inadvertent index file degradation. Refer to the Appendix, Graduate Programs Office DBMS User's Manual for DBMS troubleshooting recommendations.

Another weakness of the DBMS, and a potentially critical one, is the garbage in-garbage out factor. DBMS data integrity is dependent on accurate data entry by the user. The programmer has emphasized the criticality of accurate data entry to all current users (28, 12, 16). The programmer has suggested the database administrator establish procedures to ensure administrative personnel double check data entry, particularly with regard to grade data.

Recommendations for Follow-on Studies

This DBMS application can be expanded to provide better data management throughout AFIT. Potential areas of study, in addition to a DBMS for the Graduate Programs Office, included a DBMS to manage the thesis program, and a faculty personnel DBMS. Both of these DBMSs could be linked with the Graduate Programs Office DBMS. A study could also be initiated to develop a similar School of Engineering graduate program DBMS.

While researching registrar grade calculation procedures, the department chief expressed an interest in a follow-on study to develop data transfer procedures for student, edplan, and grade data (6). Currently, data entry is accomplished both within the Graduate Programs Office and the AFIT Registrar. This duplication of effort can be eliminated by determining responsibility for data entry and creating files transfer procedures from the AFIT mainframe computer to the Graduate Programs Office Z-248 or from the Z-248 to the AFIT mainframe computer.

To verify the data normalization process, the database administrator should provide data dictionaries to instructors teaching database management and development courses. Students could examine the DBMS data structure and identify potential problem areas not considered or overlooked by the programmer. Students in the dBASE III Plus course, LOGM490, could be provided with program coding

to evaluate the degree of efficiency achieved by the programmer and determine if DBMS programs could be made more efficient.

The Department of Defense has purchased numerous Zenith micro-computers. Many organizations are managing information manually as was the Graduate Programs Office. The need for automated database management systems to accomplish day-to-day administrative tasks is critical. In light of the potential for future defense budget cutbacks, it is essential that organizations eliminate inefficient manual information management procedures and be provided with pre-programmed automated applications such as this one. The Air Force Institute of Technology has a pool of talented, knowledgeable students who could be used to produce computer applications, both large and small. The problem is to communicate the need for organizations to provide AFIT with their application requirements and for AFIT to properly cultivate and motivate talented students who can provide these applications while fulfilling their academic curriculum requirements.

The programmer has heard student criticism of instructors for their failure to use "real-world" problems to develop student skills and apply knowledge taught in academic courses. It is suggested that instructors in computer-based courses such as statistics, quantitative decision making, and computer application courses provide course projects to students such as this one suggested by

the Graduate Programs Office or other projects suggested by organizations throughout the Department of Defense (DoD). By seeing immediate results of class projects or research activities, both students and DoD agencies would benefit.

Appendix

Graduate Programs Office DBMS

User's Manual

Graduate Programs Office DBMS User's Manual

Introduction

The Graduate Programs Office DBMS application automates the Graduate Programs Office data management functions. The DBMS uses the Ashton-Tate dBASE III Plus™ and Concentric Data Systems, Inc R&R Relational Report Writer(R) reports generator to manage data and create reports. The system allows users with basic knowledge of computer data entry processes and menu systems to input, edit, delete, and print database information. The program assumes little or no knowledge of off-the-shelf software packages used to create this application. The user can enhance his or her ability to use this package if they have an understanding of database concepts and a working knowledge of dBASE III Plus and R&R Relational Report Writer.

The Graduate Programs Office DBMS User's Manual provides a reference for DBMS operations. This manual is not a dBASE III Plus or R&R Relational Report Writer learning tool, but will guide the inexperienced user through the menu structure and assist in data entry procedures. For specific information on technical aspects of this program or for specific DBMS characteristics or program code, refer to Volume 2, Graduate Programs Office DBMS Technical Reference Manual.

DBMS Main Menu

When you enter dBASE III Plus, the first menu to appear is the Graduate Programs Office DBMS Main Menu (Figure UM-1).

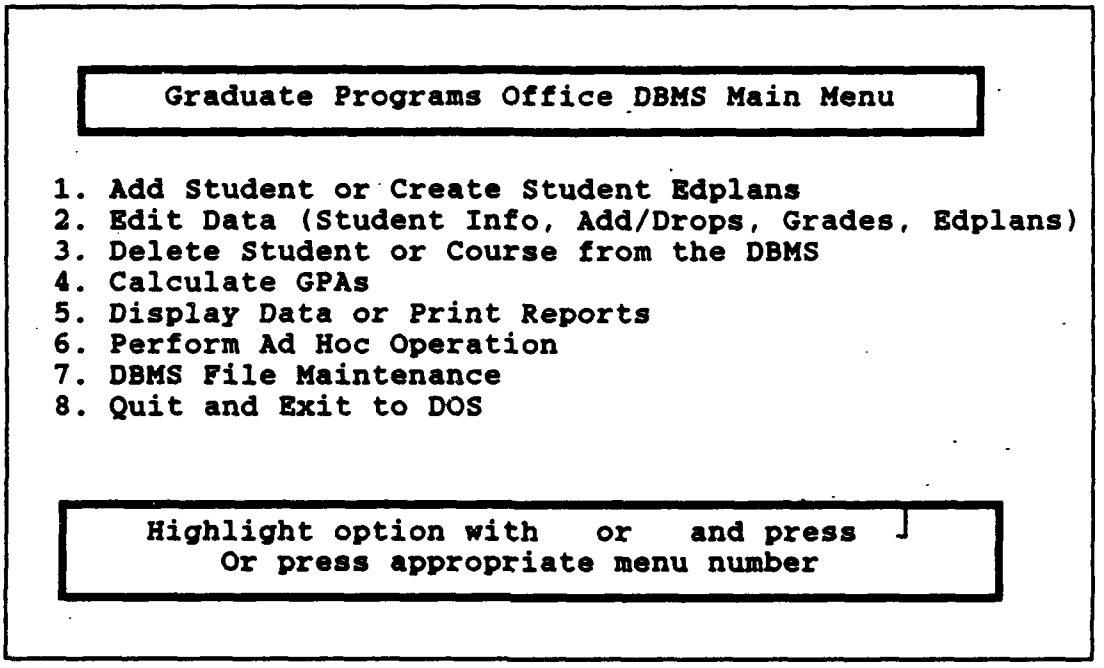


Figure UM-1. Graduate Programs Office DBMS Main Menu

You select DBMS menu options in one of two ways. You can position the lightbar (not shown in Figure UM-1) with the up or down arrows (or) or press the number corresponding to the operation you want to perform.

Option number one allows you to add student demographics data to the database or to create education plans (edplans) for an entire class or for an individual student. By selecting this option, you will proceed to the Graduate Programs Office Add Menu for additional menu option choices. See the User's Manual section on the Graduate

Programs Office Add Menu for further explanation of DBMS data entry functions.

Option number two allows you to edit existing data such as student, academic, or course data. This is the menu you will use to enter or change student grades and update edplans. By selecting this option, you will proceed to the Graduate Programs Office Main Edit Menu for further menu option choices. See the User's Manual section on the Graduate Programs Office Main Edit Menu for further explanation of DBMS edit functions.

Option number three allows you to delete all information associated with a student (demographics or academic) from the database. Selecting this option will take you to the Graduate Programs Office Delete Menu for additional menu option choices. See the User's Manual section on the Graduate Programs Office Delete Menu for further explanation of DBMS delete functions.

Option number four allows you to calculate grade point averages (GPAs) for an entire class, or a single student, in a specified quarter. Selecting this option will take you to the Graduate Programs Office Grade Calculation Menu for additional options. See the User's Manual section on the Graduate Programs Office Grade Calculation Menu for further explanation of student GPA calculation procedures.

Option number five allows you to display or print pre-defined DBMS reports. Selecting this option will take you to the Graduate Programs Office Print Menu to select

student, academic, or course related reports. See the User's Manual section on the Graduate Programs Office Print Menu for further explanation of reports.

Option number six allows you to access the dBASE dot prompt to perform ad hoc database operations. You should not use this option unless you are familiar with dBASE commands and functions. You should never use the ad hoc option to add, edit, or delete data. If you do attempt ad hoc add, edit, or delete procedures and forget to specify an index file associated with the database table you are working with, you may not be able to access that table for future operations. If you suspect you have damaged index files, see the User's Manual section on DBMS Troubleshooting. When you select option six, you will receive the following screen prompt:

```
Type DO MAINMENU from the Dot Prompt to
return to menu operations
Press any key to continue...
```

When you complete your ad hoc operations, type DO MAINMENU from the dot prompt to return to the DBMS menu system. See the User's Manual section, Ad Hoc Operations, for a discussion of ad hoc operations.

Option number seven allows you to access the Graduate Programs Office Data Save/Restore Menu. This menu contains procedures to duplicate your database files in the event something happens to the existing database data or program files. This option also allows you to remove student, academic, and grade information following a September or

December graduation. See the User's Manual section on the Graduate Programs Office Data Save/Restore Menu for further files maintenance activities information.

Option number eight allows you to exit the DBMS and return to the disk operating system (DOS). If you entered dBASE from a menu system, you will return to that menu system or to the DOS prompt, depending on your system's configuration. When you select option eight, you will receive the following prompt on the menu display:

> Do you want to quit the DBMS
> and exit to DOS?

If you enter a "Y" indicating you want to exit the DBMS, the DBMS will check to see if the last Academic Table update date is the same as the system date stored in your computer. If the dates do not match, the computer will copy all database subdirectory files from Drive C to Drive D. If the dates do match, no files are copied.

If you enter an "N" indicating you do not want to exit the DBMS, the menu screen will be re-displayed without the exit prompt.

Graduate Programs Office Add Menu

The Graduate Programs Office Add Menu is shown in Figure UM-2. You can display this menu by selecting menu option one on the Graduate Programs Office Main Menu (Figure UM-1).

Add Student to DBMS. If you want to enter student demographics information for a student who has returned his

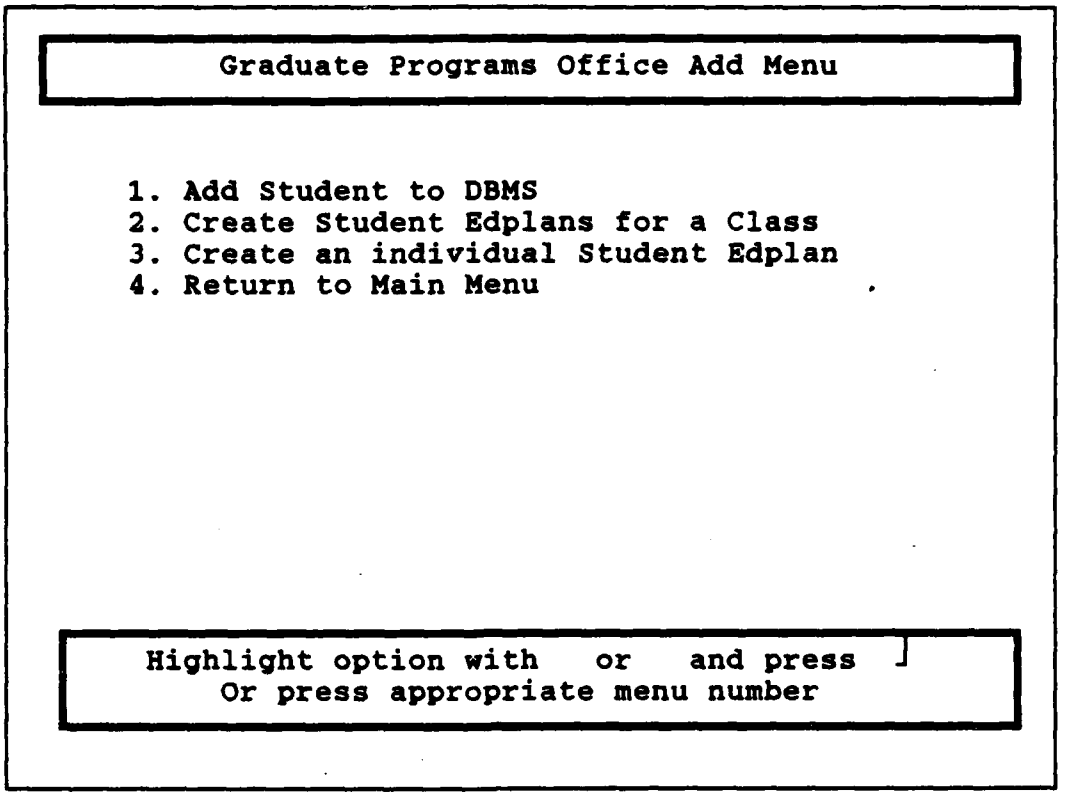


Figure UM-2. Graduate Programs Office Add Menu

or her data form, select option one. By selecting this option you will display a blank Student Data Form. The Student Data Form is seven pages of screen entry displays. The Student Data Form resembles the data form sent to students.

If you enter this form by mistake, you can exit by pressing the ESC (escape) key on the keyboard. The computer will display the message "PLEASE WAIT!!!" while it removes the blank record from the database. When the blank record has been removed, the Graduate Programs Office Add Menu will be re-displayed.

To enter student demographics data, enter the student information on the computer. Enter data using upper and lower case letters where appropriate. Where fields require a special format, the screen will display dashes (-), slashes (/). or periods (.). You do not have to enter these characters yourself. For all date fields, the format you enter must be YR/MO/DY as in 88/09/28 (28 Sep 88) or 00/11/01 (1 Nov 00). For ZIP code fields, enter a five or nine digit number. If the ZIP code only contains five numbers, press the ENTER (or return) key after entering the ZIP code to continue to the next field. You should enter the area code for all phone numbers, even those in the "513" area code.

For the SEX entry on page 1 of 7, enter a M or F only. For "Aero Rating", "Will Spouse Accompany You to AFIT", "Is Spouse Military", "If Single, Any Dependent Children", and "Will Dependent Children Accompany you Here to AFIT" entries on page 2 of 7, enter a Y or N only. For the "SIE Member" and "SIE Offered" fields on page 6 of 7, enter a Y or N only.

Cursor movement and edit commands are displayed at the bottom of each data entry screen. If you do not have data for a column, press the ENTER key and continue to the next entry. When you have entered data for the last student, press the ESC key when the cursor is in any column on page 1 of 7 of a blank student data entry screen.

If you make a mistake entering data, an error message will be displayed in the top right side of the screen. Follow the instructions shown, correct the entry, and continue. If you try to enter data and the computer will not accept keyboard entries, you have probably made an error and an error message is displayed on the top right side of the screen.

Create Student Edplans for a Class. Option two on the Graduate Programs Office Add Menu allows you to make student edplans for students in the incoming class. Before you run this routine, you should ensure all students have the correct program OPTION in the Student Table. If student options are not correct, the student's edplan will not contain the correct courses and you will have to make the changes to his or her edplan.

When you select this option, the following message will be displayed:

> Do not use this procedure if any Edplans have
> been created in the desired class!!!
> Do you want to continue?

If you are not sure whether edplans already exist enter an "N" and ask someone!!! If you are unable to determine from you co-workers if edplans have been created, ask the database administrator to check the Academic Table for student edplans. If you are sure you want to continue, enter a "Y" at the prompt.

If you continue, you will be prompted for the class number for which you want to create edplans. Enter the last

two digits for the year in which they will graduate. Do not enter the suffix (S or D). After you enter the year, the screen will display the following prompt:

Please wait, creating edplans for Class XX

The class year you entered will be displayed where the XX is shown above. As the procedure creates edplans for each student, the screen will display the person's last name, quarter, and course for the edplan being created. When the program is complete, the Graduate Programs Office Add Menu will be displayed.

Create an Individual Student Edplan. Option three on the Graduate Programs Office Add Menu will allow you to create an individual edplan for a student. Enter the student's social security number at the prompt, or make sure the entry area is blank and press the ENTER key to return to the Graduate Programs Office Add Menu.

If you enter a social security number, the computer will check to make sure the number you enter matches a student in the database. If the number is not in the Student Table, the computer will display the following prompt:

SSAN is not in the Student.DBF!!!

If the number does exist, the computer will check to see if the student already has an edplan in the Academic Table. If an edplan is found, the following prompt is displayed:

```
This student has records in the Academic.DBF!!!
Delete existing records before creating an
Edplan or use the EDIT menu to update the
Edplan!!!
Press any key to continue...
```

If you receive this prompt, you must either delete or edit existing records. If the student does not have records in the Academic Table, the computer will display the following prompt:

```
Please wait, creating a student edplan!
```

When the program is complete, the Graduate Programs Office Add Menu will be re-displayed.

Return to Main Menu. Option four allows you to return to the Graduate Programs Office Main Menu.

Graduate Programs Office Main Edit Menu

The Graduate Programs Office Main Edit Menu is shown in Figure UM-3. This menu is accessed by selecting menu option two from the Graduate Programs Office DBMS Main Menu (Figure UM-1).

Edit Student Demographic Information. Option one on the Graduate Programs Office Main Edit Menu allows you to edit student demographic information. Selecting this option will display the Graduate Programs Office Student Edit Menu (Figure UM-4).

Edit Name, SSAN, Class, or Option Only. Option one on the Graduate Programs Office Student Edit Menu allows

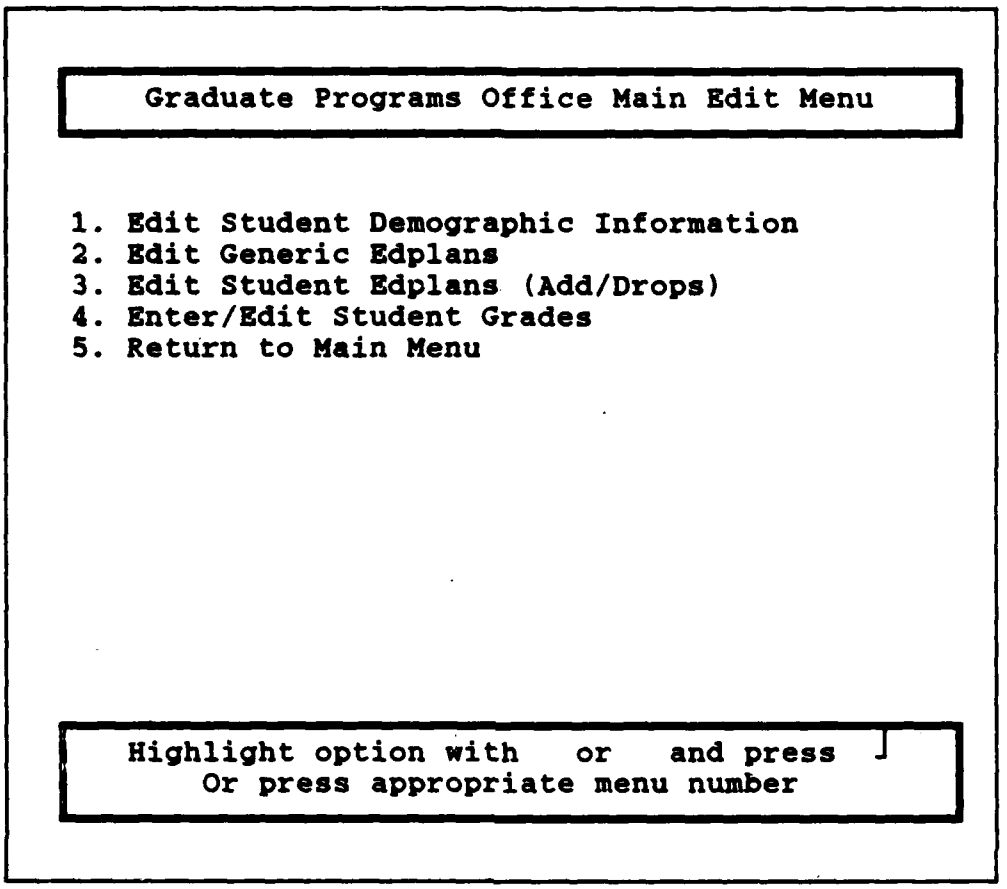


Figure UM-3. Graduate Programs Office Main Edit Menu

you to edit a student's name, social security number, class designator, or program option only. Selecting this option will display a menu that allows you to change any one data column at a time. In each case, you will be prompted to enter the social security number for the student whose data you will be changing. After the data is updated by the computer you will return to the menu. Select option five to return to the Graduate Programs Office Student Edit Menu.

Edit Any Information in a Student Record. Option two on the Graduate Programs Office Student Edit Menu allows

you to edit any student demographics information except the name, SSAN, Class, or Option. The screen format editing

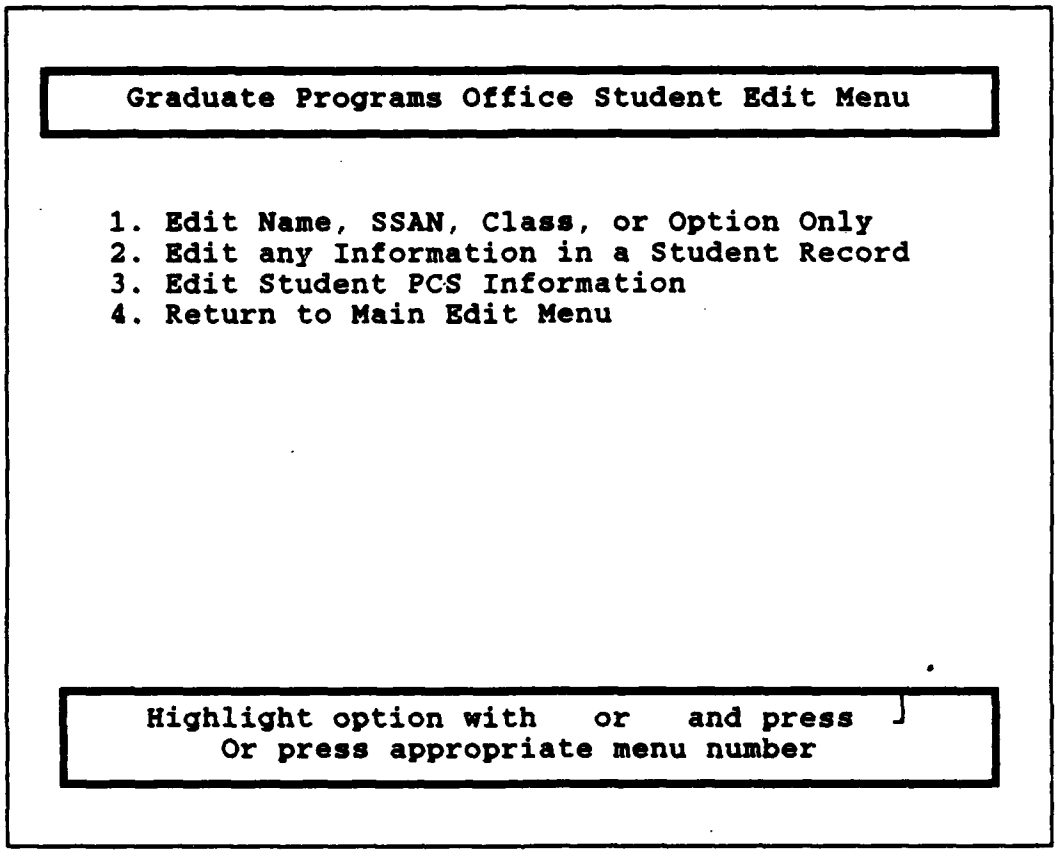


Figure UM-4. Graduate Programs Office Student Edit Menu

demographics data is the same format used when entering student demographics data. Access to the name, SSAN, Class, and Option columns is not allowed by the screen format. All other information in the student demographics record can be edited by moving the cursor to the appropriate column using the PgUp or PgDn keys to move from screen to screen or using the up or down arrows (or) to move from column to column on the same screen.

Edit Student PCS Information. Option three on the Graduate Programs Office Student Edit Menu allows you to enter or edit only the screen containing information related to the student's next assignment or forwarding address following graduation. Edit functions for this screen are the same as with the larger student demographics data screens.

Return to Main Edit Menu. Option four on the Graduate Programs Office Student Edit Menu allows you to return to the Graduate Programs Office Main Edit Menu.

Edit Generic Edplans. Option three on the Graduate Programs Office Main Edit Menu allows you to edit any of the edplans prepared by the various program option managers. Use this menu option to make edplan changes prior to creating student edplans for a new class. If you do not update edplans before you create student edplans, you will create a considerable amount of editing because student edplans for a new class will be incorrect for that program option.

Select the program option you want to edit by entering the number that corresponds to the edplan you want to change. If you want to change a course, position the cursor in the course column and type the new course designator over the old course designator. If you want to delete a course, position the cursor in any column in that row of the table and press the Ctrl and "U" keys to mark the record for deletion. To add a course, move the cursor to the last row

in the table and press the down ( ) arrow. The computer will display the message below the status line "===> Add new records? (Y/N)."

Press the "Y" key and enter the quarter, course designator, and number of hours. When you have entered the data, press the up ( ) arrow and the computer will automatically position the course by quarter and alphabetically within the edplan. Press Ctrl and End keys simultaneously to save the changes or press the ESC key to abort the edit process. The screen will display the prompt:

Do you want to edit another edplan? (Y/N)

If you enter "Y" for yes, the list of program options will be re-displayed for your selection. If you enter "N" for no, you will return to the Graduate Programs Office Main Edit Menu.

Edit Student Edplans (Add/Drops). Option three on the Graduate Programs Office Main Edit Menu allows you to enter add/drop data. When you select this option the computer will display a message asking you to enter the social security number for the student whose edplan you want to edit or press the ENTER key, if the column is blank, to return to the Graduate Programs Office Main Edit Menu. If you enter a valid student social security number, you will be prompted to enter a number corresponding to the academic quarter you want to do the add/drop. If you want to enter AFIT courses the student entered the program with, specify

zero (0), the "Entering Course Credit" quarter. When you specify the quarter, the computer will display a list of all courses the student has for that quarter. If you entered zero for the quarter and had not previously entered courses, the list will not have any entries. Below the list of courses, the computer will display the following list of options:

```
1 - Change Course in List to New Course
2 - Add Course to List
3 - Delete Course from List
4 - Exit and abort edit

Enter Option Number:
```

If you specify option one, you will be prompted to enter the course number you want to change. From the list displayed, enter the course number you want to change. The computer will then prompt you to enter the new course number, section number (if known), and the number of hours. A new list of courses will be printed for that quarter and the computer will prompt you to determine if the change you made was correct. If you enter a "Y" for yes, the computer will ask you if you want to do another add/drop for the same student. If you enter an "N" for no, the computer will re-display the list and you will be allowed to change, add, or delete the list again.

If you want to do another add/drop for the same student, enter a "Y" at the prompt. You will be prompted to specify another quarter to edit. If you do not want to do

another add/drop for the same student, enter an "N" at the prompt. You will be asked if you want to edit another student's edplan. If you enter a "Y" for yes, the computer will take you back to the edit procedure, if you enter an "N" for no, the computer will return you to the Graduate Programs Office Main Edit Menu.

If you are entering add/drop data and you specify a course that is not in the Course Table, the computer will ask you to determine if your entry was valid. If you did not make a typing mistake and are sure you are entering a valid course designator, you will be prompted to enter the course data in the Course Table. The computer will prompt you to enter the course name, number of hours, and a true/false entry for graduate course status. If the course is a graduate level course, enter a "T". If it is not a graduate level course, enter a "F". When you have entered the new course data, the add/drop edit procedure described in the previous paragraphs will resume.

Enter/Edit Student Grades. Option four on the Graduate Programs Office Main Edit Menu allows you to enter or change an individual student's grade or enter grades by course and section at the end of each academic quarter. When you specify this menu option, the Graduate Programs Office Student Grade Edit Menu will be displayed (Figure UM-5).

Enter Grades for a Specified Course and Section. Option one on the Graduate Programs Office Student Grade Edit Menu allows you to enter end-of-quarter grades for a

specific course and section. You will be prompted to enter the number of the next graduating class, the academic quarter, and course name, and the course section. The

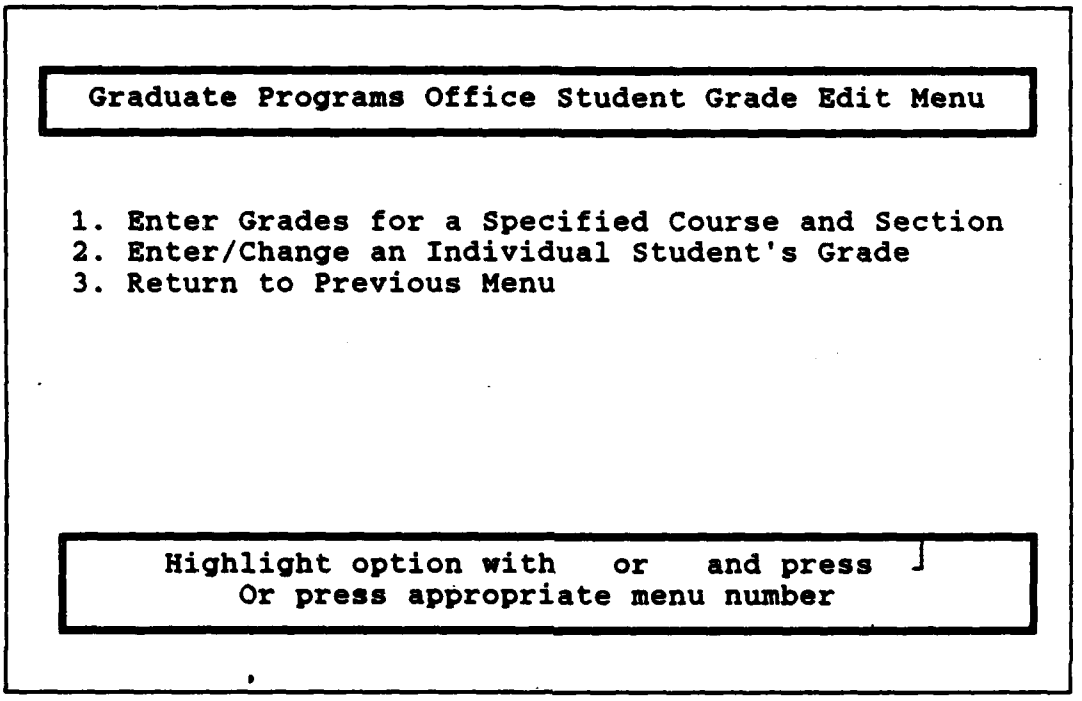


Figure UM-5. Graduate Programs Office Student Grade Edit Menu

computer will search the Academic Table for students enrolled in that course and section and allow you to enter the grade using the display shown in Figure UM-6. You can only enter valid grades listed in the Graduate Programs Handbook. If you enter an illegal grade, the computer will display an error message and allow you to re-enter a valid grade. The computer will then ask you if this is the last student in the section. If it is the last student or if you cannot continue entering grades, enter a "Y" and the computer will ask you if you want to enter more course

grades. If you enter a "Y" indicating you want to continue to enter course grades, you will be prompted for the year, quarter, course, and section as before. If you do not want to enter additional course grades, you will be returned to the Graduate Programs Office Student Grade Edit Menu.

Graduate Programs Office Grade Posting Display				
Last Name:			First Name:	
SSAN:			Class:	
Quarter:	Course:	Section:	Grade:	Hours:

Figure UM-6. Graduate Programs Office Grade Posting Display

Enter/Change an Individual Student's Grade. If you select option two from the Graduate Programs Office Student Grade Edit Menu, you will be prompted to enter the social security number for the student whose grades you will edit or press ENTER with a blank social security number to return to the previous edit menu.

The procedure for grade editing is very similar to add/drop editing. You will be asked to enter the quarter and course for which you want to edit grades. The old grade will be displayed and you will be prompted to enter the new grade. The computer will ask you if the information you entered is correct. If it is and you enter a "Y", you will be asked if you want to edit grades for another individual.

If you want to edit other students' grades, enter a "Y" for yes and you will be prompted for the student's social security number, the quarter to edit grades, and the course number as before. If you do not want to enter or edit more student grades, enter an "N" when asked if you want to edit more grades and you will return to the Graduate Programs Office Student Grade Edit Menu.

Return to Previous Menu. Option three on the Graduate Programs Office Student Grade Edit Menu allows you to return to the Graduate Programs Office Main Edit Menu.

Return to Main Menu. Option five on the Graduate Programs Office Main Edit Menu allows you to return to the Graduate Programs Office DBMS Main Menu.

Delete Student or Course from the DBMS

If a student is disenrolled from a graduate program, you will want to delete all information for that student from the database. If a course is no longer part of the curriculum and no systems or logistics student has taken that course during the current year, you will want to delete that course from the Course Table. Select option three from the Graduate Programs Office DBMS Main Menu to get to the Graduate Programs Office Delete Menu. The Graduate Programs Office Delete Menu is shown in Figure UM-7.

Delete Student from DBMS. Selecting option one from the Graduate Programs Office Delete Menu allows you to delete all student demographics, academic, and grade data

for that student from the database. You will be prompted to enter the social security number for the student whose data you want to delete. If you enter a valid social security number, the computer will ask you if you want to delete

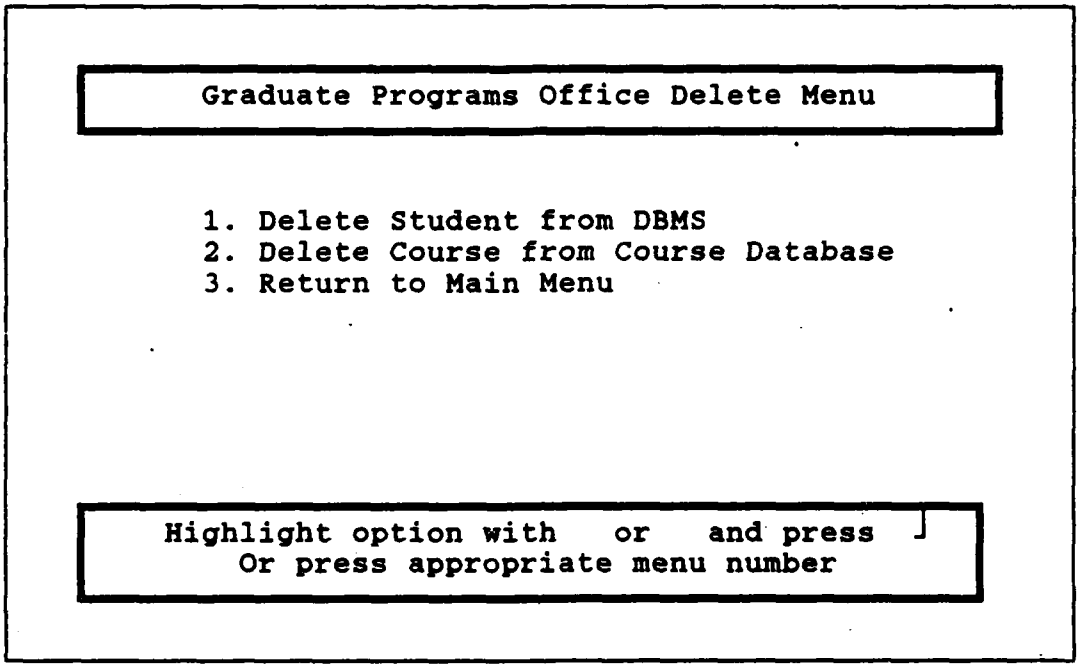


Figure UM-7. Graduate Programs Office Delete Menu

another student from the DBMS. If you enter a "Y", you will be asked to enter another social security number for the next student who is being deleted from the database. When you enter an "N" after deleting student data, the computer will actually delete the data from the database. When the computer finishes deleting student data, the Graduate Programs Office Delete Menu will be re-displayed.

Delete Course from Course Database. If you select option two from the Graduate Programs Office Delete Menu, the computer will ask you to enter the course you want to

delete from the DBMS. If the course is found in the Course Table, it will be deleted and you will be asked if you want to delete another course. If the course is not in the Course Table, the computer will advise you and ask you if you want to delete another course. When you enter an "N" to the question of deleting another course, the computer re-displays the Graduate Programs Office Delete Menu.

Return to Main Menu. Selecting option three from the Graduate Programs Office Delete Menu returns you to the Graduate Programs Office DBMS Main Menu.

Calculate GPAs

To calculate GPAs for a student or an entire student class, select option four from the Graduate Programs Office DBMS Main Menu. Selecting this option takes you to the Graduate Programs Office Grade Calculation Menu shown in Figure UM-8.

Calculate GPA by Program Year. This option allows you to compute grades for an entire class by academic quarter. When you select this option, you are prompted to enter the academic quarter and the class designator for the student class whose grades you wish to calculate. Because this procedure takes a long time and cannot be easily halted once it begins, the computer asks you if you have specified the right class and quarter criteria for the grade calculation. If you enter a "Y" for yes, the calculation routine continues, otherwise, you will return to the Graduate

Programs Office Grade Calculation Menu. Next, the computer checks to see if any records meet the class and quarter criteria you specified and if so, calculates grades for that class and quarter. This procedure takes approximately 30 minutes and should not be done once the initial grades have been computed. If you only want to calculate grades for a few individuals, you should use option two from the Graduate Programs Office Grade Calculation Menu.

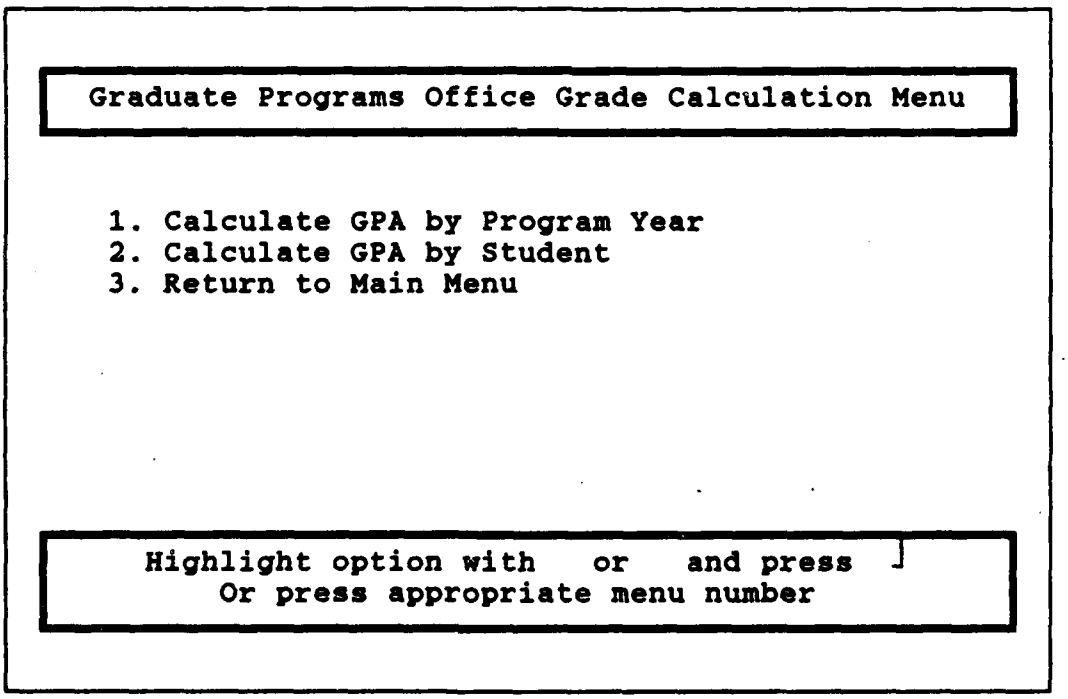


Figure UM-8. Graduate Programs Office Grade Calculation Menu

Calculate GPA by Student. This option is similar to the option described for calculating GPAs by program year. You will be prompted to enter the academic quarter, the student's class year, and the student's social security number. If the criteria you specify is valid, and records

that meet that criteria exist, the computer continues with student grade calculation. Otherwise, you will be returned to the Graduate Programs Office Grade Calculation Menu. When the process is complete, the Graduate Programs Office Grade Calculation Menu is re-displayed.

With both grade calculation options, you can only calculate one quarter's grades at a time. If you change a student's grade after you have calculated the GPAs initially, you will have to recalculate GPAs for each quarter starting with the quarter where a grade was first changed and ending with the quarter whose grades were last entered in the database.

Return to Main Menu. Option three on the Graduate Programs Office Grade Calculation Menu allows the user to return to the Graduate Programs Office DBMS Main Menu.

Display Data or Print Reports

Option five on the Graduate Programs Office DBMS Main Menu allows you to display or print pre-programmed reports that you will most often need. Figure UM-9 shows the Graduate Programs Office Print Menu. The menu divides the 61 pre-programmed reports into three sub-menus. Reports are organized by the type of data they contain: student, academic, or course data.

Display or Print Student Information. Selecting option one from the Graduate Programs Office Print Menu allows you to display or print student data reports for a specified

class. When you select this option, you may display or print one of the sixteen reports shown in Figure UM-10 or return to the Graduate Programs Office Print Menu. When you enter a report number, the computer will prompt you to enter a class year and ask you if you want to print the report. If you enter a "Y" indicating you want to print the report, the report output will be sent to the printer. Make sure

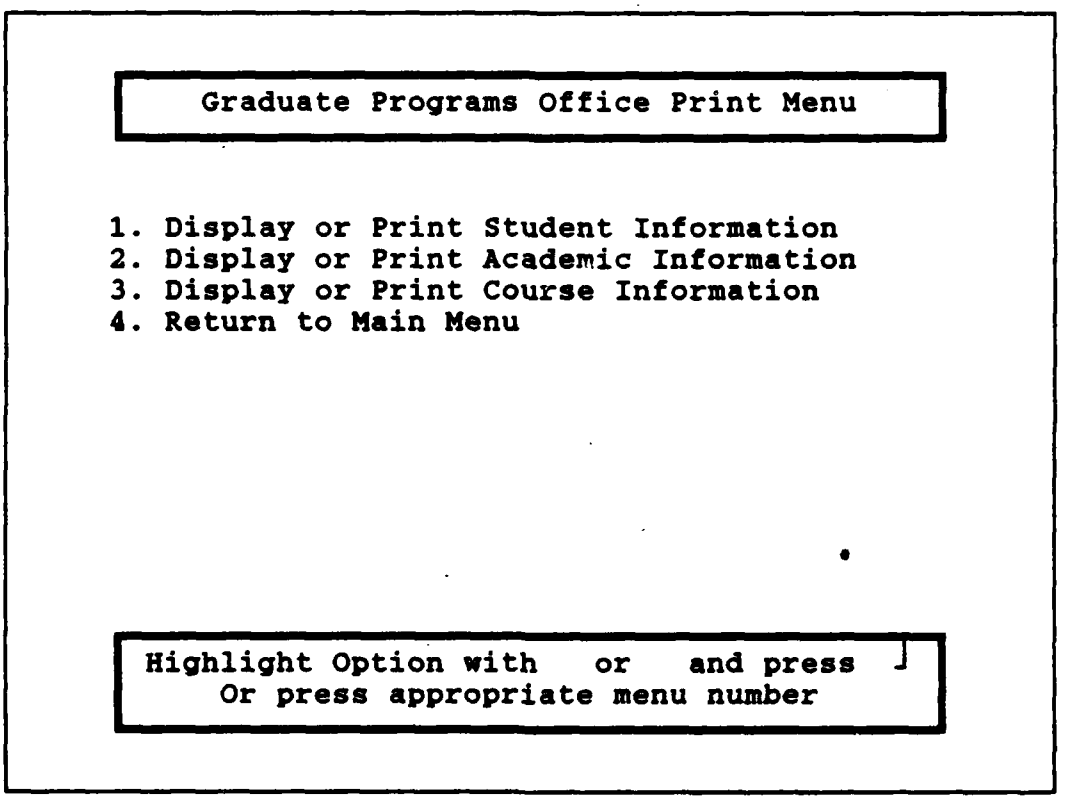


Figure UM-9. Graduate Programs Office Print Menu

the printer is turned on!!! If you enter an "N" at this prompt, the report output will be displayed on the computer screen.

When processing reports, the report generator requires a few minutes to load and sort data. If you entered an "N"

for report printing, the first screen of data will be displayed on the computer screen and a data display select line will appear at the bottom of the screen. You may display the rest of report by pressing the "L" key to advance the display a line at a time, the "S" key to advance the display a screen at a time, a "C" to continuously display report data, an "R" to restart the display process, or a "Q" to quit the report display and return to the Student Data Reports display.

```
                    STUDENT DATA REPORTS

1 - Academy Graduates          9 - Alpha Roster
2 - Civilian Students          10 - Class Section Roster
3 - International Students     11 - Date of Rank Roster
4 - Male/Female                12 - Command & Base
5 - Married Students           13 - Photo Book
6 - Single Students            14 - SIE Membership
7 - Rated Students             15 - SIE Eligibles
8 - SSAN Roster                16 - Student Demographics

                  17 - Return to Print Menu

 Enter the number for the report you want to print:
```

Figure UM-10. Student Data Reports Display

If you entered a "Y" to print the report, the report will print when data is loaded and sorted. When report printing

is complete, the Student Data Reports Display will be re-displayed.

Display or Print Academic Information. Option two on the Graduate Programs Office Print Menu allows you to select academic data reports for display or printer output. Figure UM-11 shows the Academic Data Reports Display.

The procedure for selecting and displaying or printing reports is similar to the process described when printing student data reports. With some reports such as the IUDF Grades Reports and Transcripts, the computer may require as

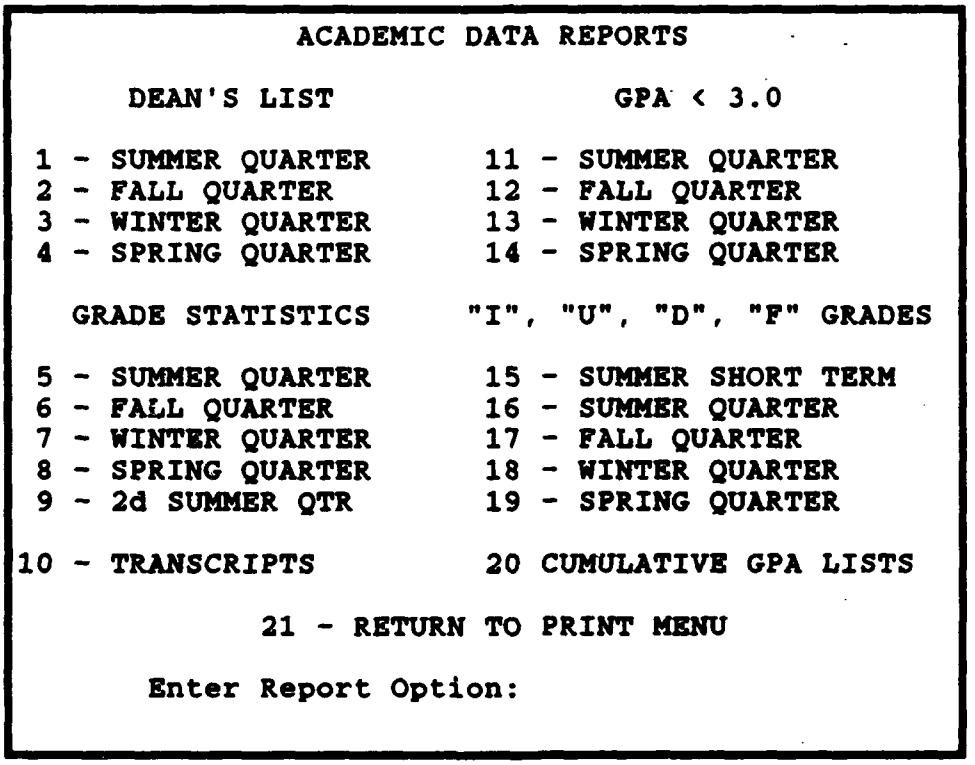


Figure UM-11. Academic Data Reports Display

much as 30 minutes to load and sort data due to the large data in the Academic Table.

Display or Print Course Information. Option three on the Graduate Programs Office Print Menu allows you to select course data reports for display or printer output. Figure UM-12 shows the Course Data Reports Display.

As with academic data reports, some reports will require time to prepare. Course Loads Reports will require approximately 40 minutes as the computer also determines course loads in addition to displaying or printing the result. To display or print reports, respond to the computer prompts as described in previous sections of this manual.

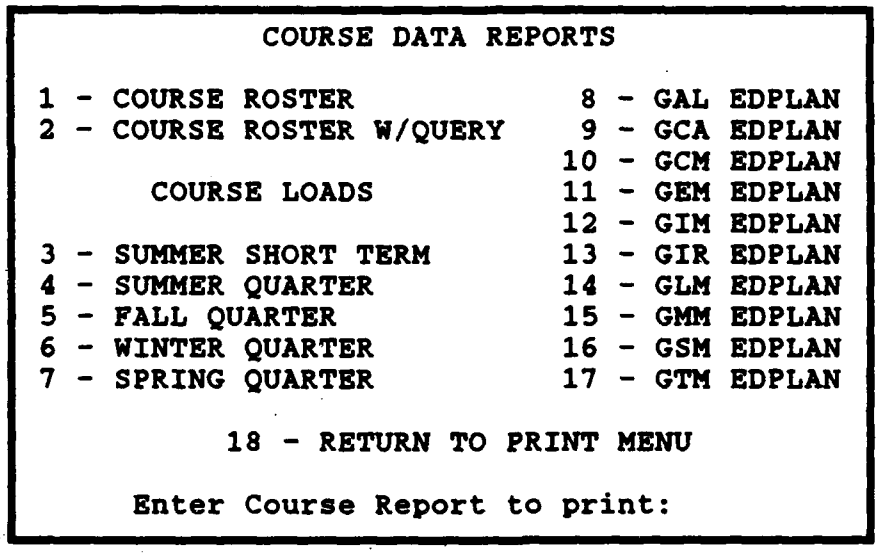


Figure UM-12. Course Data Reports Display

Return to Main Menu. Select option four from the Graduate Programs Office Print Menu to return to the Graduate Programs Office DBMS Main Menu.

Perform Ad Hoc Operation

This option, number six, on the Graduate Programs Office DBMS Main Menu allows the experienced dBASE III Plus user to perform ad hoc data queries. You should use this option to execute dBASE "LIST", "COUNT", or "AVERAGE" commands. Do not use this option to perform ad hoc data adding, editing, or deleting. You may damage index files if you do not specify all related index files while performing these operations.

DBMS File Maintenance

Select option seven from the Graduate Programs Office DBMS Main Menu if you want to backup or save data files. Obtain the floppy disks used for external data saves from the database administrator. The Graduate Programs Office Data Save/Restore Menu is shown in Figure UM-13.

Back Up Files From Drive C to Drive D. Option one on the Graduate Programs Office Data Save/Restore Menu allows you to copy all files in the DBASE subdirectory on Drive C to the DBASE subdirectory on Drive D. The procedure takes approximately five minutes. Once you select this option, no further inputs are required and the Graduate Programs Office Data Save/Restore Menu will be re-displayed when all files have been copied.

Back Up Files from Drive C to Floppy Disks. Option two on the Graduate Programs Office Data Save/Restore Menu allows you to save DBMS program, data, and index files on

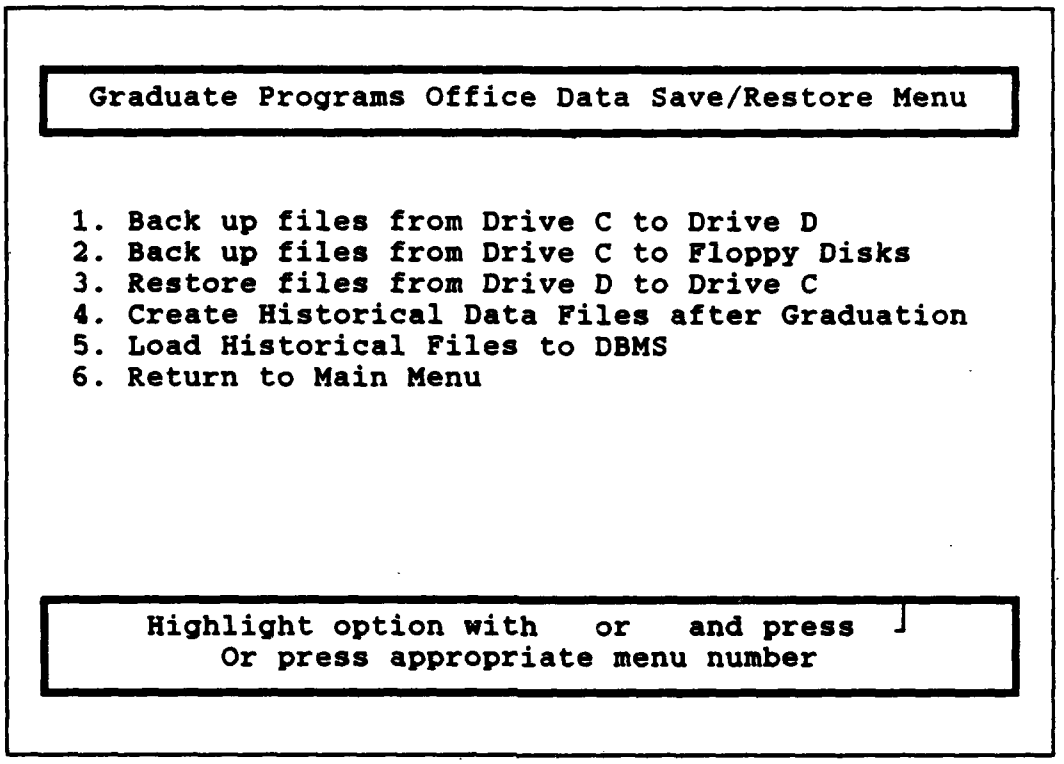


Figure UM-13. Graduate Programs Office Data Save/Restore Menu

floppy disks. Data can be restored to the hard disk, Drive C, if data is lost or files are damaged. You must have the nine 5.25 inch DBMS floppy disks to perform this procedure. Obtain these disks from the database administrator. When you select this option, the computer will prompt you to change disks. Do not remove a disk while the red drive active indicator is on. Wait until the indicator goes out before removing the disk.

Restore Files from Drive D to Drive C. Option three on the Graduate Programs Office Data Save/Restore Menu allows you to copy files from Drive D back to Drive C. You would use this option if data files on Drive C were damaged or

erased. When you select this option, no further user inputs are required and the Graduate Programs Office Data Save/Restore Menu will be re-displayed when all files have been copied.

Create Historical Data Files After Graduation. Option four on the Graduate Programs Office Data Save/Restore Menu allows you to remove student demographic, academic, and grade data from the DBMS and transfer it to floppy disk for archiving. If you are creating historical files following a September graduation, you must have three FORMATTED floppy disks. If you are adding December graduates to the disks with the September graduates, obtain the appropriate class disks prior to starting this procedure. When you select this option, the computer will prompt you to enter the class year for the graduates and insert a disk into Drive A when directed.

Load Historical Files to DBMS. Option five on the Graduate Programs Office Data Save/Restore Menu allows you to load student demographic, academic, and grade information by class year. The historical data is added to existing DBMS data files and will be organized within the respective data files as it is loaded. This procedure should be used if existing file structures will be modified. If historical data files are not kept current structurally, they will be difficult to combine with files having a different structure.

Return to Main Menu. Option six on the Graduate Programs Office Data Save/Restore Menu allows you to return to the Graduate Programs Office DBMS Main Menu.

DBMS Troubleshooting Procedures

If you encounter problems with the DBMS, notify the database administrator immediately. Note any error message you receive to assist in debugging the problem. There are two potential problems that might be encountered: data searches that should provide information but do not provide data or, program interrupts.

If you attempt a database function or ad hoc query that should result in a successful database search but no records are listed, attempt to solve the problem as follows:

1. If the problem is with a menu function, notify the database administrator of the problem. The database administrator will need to know what procedure you were trying to perform and what happened when you tried. Note all error messages to assist in program debugging. The problem may be damaged index files. The database administrator should use the ad hoc data menu option to access the dBASE Assist Program. Select the database file using the Assist Program and select all related index files. Use the ESC key to return to the dot prompt and type the command "REINDEX". When files are reindexed and you return to the dot prompt, type "DO MAINMENU" and reattempt the procedure that did not work correctly. If further problems are encountered, the database administrator will have to analyze error messages to determine the cause of the problem.

2. If a problem occurs during an ad hoc operation, chances are you have not entered the command correctly, or you have not properly specified a query condition. When performing data queries, the sequencing of ".AND." or ".OR." is critical to achieving a proper search. Ensure

parentheses are used, if required, to remove any query ambiguities. Failure to use parentheses when required can lead to data search failure.

If you accidentally press the ESC key, you may cause a program to halt. If this occurs a message will appear on the screen:

```
*** INTERRUPTED ***
Called from - C:
Called from - C:
Cancel, Ignore, Suspend? (C, I, or S)
```

If you caused the problem, type "C" for cancel and type "DO MAINMENU" at the dot prompt to resume menu operations or continue from the dot prompt for ad hoc operations.

If a program error occurs, the INTERRUPTED message will also appear. This indicates a major programming error and the database administrator should be notified. You should note any error messages to aid in program debugging. If you want to continue, enter a "C" for cancel, and reattempt the operation where the error first occurred. An error in one DBMS program does not necessarily affect other DBMS programs. Other DBMS programs should still function properly.

For specific information on technical aspects of this program or for specific DBMS characteristics or program code, refer to Volume 2, Graduate Programs Office DBMS Technical Reference Manual.

Bibliography

1. Atre, S. Data Base: Structured Techniques for Design, Performance, and Management. New York: John Wiley & Sons, 1980.

2. Brietling, Lt Cmdr James D., Supply Requirements Officer, Polaris Missile Office, Atlantic. Telephone Interview. PMOLANT, Charleston Naval Base SC, 18 August 1988.

3. Buck, Captain Stephen D. A Database Management System to Manipulate Data Collected at the National Training Center, Ft. Irwin CA.. MS thesis. Naval Postgraduate School, Monterey CA, June 1987 (AD-A183197).

4. Collins, C. Francis, Computer Assistant, AFIT/RRDS. Personal Interview. Air Force Institute of Technology, Wright-Patterson Air Force Base OH, 24 August 1988.

5. Concentric Data Systems, Inc. R&R Relational Report Writer, Version 3. Westborough MA: Concentric Data Systems, Inc, 1988.

6. Curtis, 1 Lt Brent G., Chief, Academic Scheduling and Registration, AFIT/RRDS. Personal Interview. Air Force Institute of Technology, Wright-Patterson Air Force Base OH, 24 August 1988.

7. Daft, Richard L. and Richard M. Steers. Organizations: A Micro/Macro Approach. Glenview IL: Scott, Foresman and Company, 1986.

8. Department of the Air Force. AFIT Catalog, 1987-1989. Montgomery AL: Government Printing Office, 1987.

9. Department of the Air Force. Air Force Institute of Technology. AUR 23-3. Montgomery AL: Government Printing Office, 1987.

10. Department of the Air Force. Unit Effectiveness Inspection Report, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 1-10 December 1987. Montgomery AL: Government Printing Office, December 1987.

11. Donahue, Bernard A. and others. Managing Underground Storage Tank Data Using dBASE III Plus. USA-CERL Technical Report N-87/21. Champaign IL: Construction Engineering Research Laboratory, 1987.

12. Gardner, Sgt Donald E. Jr., NCOIC, Graduate Student Services. Personal Interviews. Air Force Institute of Technology, Wright-Patterson Air Force Base OH, 7 October 1987 through 22 August 1988.

13. Goley, George F., IV. Microcomputer Database Management Systems. Homewood IL: Dow Jones-Irwin, 1987.

14. Halliday, Lt Col John M., Head, Department of Logistics Management. Personal Interviews. Air Force Institute of Technology, Wright-Patterson Air Force Base OH, 30 September 1987 through 24 February 1988.

15. House, William C. Interactive Decision Oriented Data Base Systems. New York: Petrocelli/Charter, 1977.

16. Henry, George R., Educational Clerical Assistant. Personal Interviews. Air Force Institute of Technology, Wright-Patterson Air Force Base OH, 15 March 1988 through 26 August 1988.

17. Isett, Major John B., Assistant Professor of Information Systems. Telephone Interview. Naval Postgraduate School, Monterey CA, 18 August 1988.

18. Kim, Major Sam Nam and Captain Jae Bock Park. Application of a Database System for Korean Military Personnel Management. MS thesis. Naval Postgraduate School, Monterey CA, March 1987 (AD-A181663).

19. Kroenke, David M. and Kathleen A. Dolan. Database Processing: Fundamentals, Design, Implementation. Chicago: Science Research Associates, Inc., 1988.

20. Martin, James. Principles of Data-Base Management. Englewood Cliffs NJ: Prentice-Hall, Inc., 1976.

21. McBride, Lt Col Dorothy J., Instructor of Systems Management. Personal Interviews. Air Force Institute of Technology, Wright-Patterson Air Force Base OH, 23 August 1988 through 25 August 1988.

22. Mikucki, Walter, Team Leader, Environmental Engineering Team. Telephone Interview. U.S. Army Construction Engineering Laboratory, Champaign IL, 22 August 1988.

23. Peschke, Lt Col Robert E., Head, Department of Quantitative Management. Personal Interviews. Air Force Institute of Technology, Wright-Patterson Air Force Base OH, 2 October 1987 through 24 February 1988.

24. Pollis, Gerald M., Environmental Engineer. Telephone Interview. Environmental and Natural Resource Division, Headquarters, U.S. Army Training and Doctrine Command, Fort Monroe VA, 22 August 1988.

25. Pratt, Philip J. Microcomputer Database Management Using dBASE III Plus. Boston: Boyd & Fraser Publishing Company, 1988.

26. Simpson, Alan. dBASE III Plus™ Programmer's Reference Guide. San Francisco: Sybex, 1987.

27. Smith, Lt Cmdr Claire C. Supply Hotlist Report Generation for Fleet Ballistic Missile Submarine Management Meetings. MS thesis, AFIT/GLM/LSM/87S-70. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson Air Force Base OH, September 1987 (AD-A186677).

28. Westfall, Lt Col Frederick W., Graduate Programs Director. Personal Interviews. Air Force Institute of Technology, Wright-Patterson Air Force Base OH, 1 October 1987 through 26 August 1988.

VITA

Captain Phillip H. Beard [redacted] in 1974 [redacted] attended the University of Texas at Austin, from which he received the degree of Bachelor of Business Administration in Transportation in 1978. Upon graduation, he was commissioned in the USAF through the ROTC program. He began his Air Force career as a Minuteman III missile combat crew member at Minot Air Force Base, North Dakota in February 1979. He then served as a Ground Launched Cruise Missile Instructor, Executive Support Officer, and Headquarters Section Commander at the 868th Tactical Missile Training Group at Davis-Monthan Air Force Base, Arizona, from October 1983 until entering the School of Systems and Logistics, Air Force Institute of Technology, in May 1987.

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GLM/LSG/88S-3

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Systems and Logistics

6b. OFFICE SYMBOL (If applicable)
AFIT/LSM

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State, and ZIP Code)
Air Force Institute of Technology
Wright-Patterson AFB OH 45433-6583

7b. ADDRESS (City, State, and ZIP Code)

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
See Box 19

12. PERSONAL AUTHOR(S)
Phillip H. Beard, B.B.A., Captain, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
MS Thesis	FROM _____ TO _____	1988 September	101

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Database Management Systems, Data Bases, Computer Programming, Personnel Management, Students
12	05		
05	09		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: A DATABASE MANAGEMENT SYSTEM APPLICATION
FOR THE GRADUATE PROGRAMS OFFICE
OF THE SCHOOL OF SYSTEMS AND LOGISTICS
VOLUME 1: DEVELOPMENT AND USER'S MANUAL

Thesis Chairman: Frederick W. Westfall, Lt Col, USAF
Graduate Programs Director

Approved for public release IAW AFR 190-1.

WILLIAM A. MAJER
Associate Dean
School of Systems and Logistics
Air Force Institute of Technology (AU)
Wright-Patterson AFB OH 45433

17 Oct 88

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
Frederick W. Westfall, Lt Col, USAF	(513) 255-4149	AFIT/LSG

thesis

The purpose of this ~~study~~ was to develop an efficient computer-based database management system ~~(DBMS)~~ application to automate manual information processing procedures used by the Air Force Institute of Technology School of Systems and Logistics Graduate Programs Office.

The author was able to create an efficient DBMS application that met the needs of the Graduate Programs Office using the Ashton-Tate dBASE III Plus(TM) DBMS and the Concentric Data Systems R&R Relational Report Writer(R). The application was implemented upon completion.

~~Volume 1 contains four chapters and an appendix.~~ The ~~Chapter I,~~ Introduction provides background on the AFIT Graduate Programs Office and their automated data processing requirements, examines characteristics of good DBMSs, examines DBMS development lifecycle, discusses software selection criteria, and examines four DBMS applications developed in 1987. Chapter II ~~Methodology,~~ documents the methodology used in developing the DBMS. Chapter III ~~Findings and Analysis,~~ discusses the programmer's incorporation of good DBMS characteristics presented in Chapter I and discusses whether the author was successful in achieving his goal of solving the specific problem. Chapter IV ~~Conclusions and Recommendations,~~ describes the impact on Graduate Programs Office operations using the DBMS and recommends follow-on studies. The appendix, Graduate Programs Office User's Manual, describes DBMS Operations. (KR)